



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

**NÁVRH EXPERIMENTÁLNÍHO ROBOTICKÉHO
PRACOVISTĚ PRO MANIPULAČNÍ OPERACE**

DESIGN OF AN EXPERIMENTAL ROBOTIC CELL FOR MANIPULATION OPERATIONS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Martin Bednář

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Aleš Pochylý

BRNO 2018

Zadání diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Student: **Bc. Martin Bednář**
Studijní program: Strojní inženýrství
Studijní obor: Výrobní stroje, systémy a roboty
Vedoucí práce: **Ing. Aleš Pochylý**
Akademický rok: 2017/18

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Návrh experimentálního robotického pracoviště pro manipulační operace

Stručná charakteristika problematiky úkolu:

Cílem práce je návrh experimentálního pracoviště s průmyslovými roboty pro manipulační operace. Práce se zabývá integrací dílčích systémů, resp. návrhem layoutu pracoviště, tvorbou simulačního modelu v SW Siemens Process Simulate včetně virtuálního zprovoznění.

Cíle diplomové práce:

1. Návrh koncepce pracoviště pro daný typ aplikace.
2. Integrace dílčích systémů pracoviště, resp. vytvoření modelu pracovní buňky.
3. Vytvoření simulačního modelu pracoviště v SW Siemens Process Simulate.
4. Virtuální zprovoznění robotické buňky.

Seznam doporučené literatury:

SICILIANO, B. KHATIB, O. Springer Handbook of Robotics. Springer-Verlag New York, Inc., 2008. 1611 s. ISBN 978-3-540-23957-4

PIRES, J. N. Industrial Robots Programming: Building Applications for the Factories of the Future. Springer, 2008. 282 s. ISBN 978-0-387-23325-3

NOF, S. Y. Springer Handbook of Automation. Springer, 2009. 1812 s. ISBN 978-3-540-78830-0

MONKMAN, G. J., HESSE, S., STEINMANN, R. SCHUNK, H. Robot Grippers. Wiley-VCH Verlag, 2007. 463 s. ISBN 978-3527406197

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2017/18

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Diplomová práce se zabývá návrhem experimentálního pracoviště s průmyslovými roboty pro manipulační operace, jež bude sloužit pro edukativní a demonstrační účely. Práce se zabývá integrací dílčích systémů, resp. návrhem layoutu pracoviště. Klíčovou fází je tzv. virtuální zprovoznění. Unikátní řešení, jež je obsaženo v diplomové práci, kombinuje při návrhu robotického pracoviště více softwarových nástrojů. Statický 3D model vytvořený v Autodesk Inventor Professional se importuje do prostředí Tecnomatix Process Simulate, kde se tak vytváří 3D model dynamický, resp. simulační model. Řídící prvek je softPLC společnosti Beckhoff vytvořené v prostředí TwinCAT. Následné propojení simulačního modelu a řídicího členu je realizováno prostřednictvím standardizovaného komunikačního protokolu OPC UA. Ukázalo se, že tato kombinace může pro praxi představovat levný nástroj pro virtuální zprovoznění, jelikož je zapotřebí pouze jednoho osobního počítače.

ABSTRACT

Final thesis is focused on design of experimental workplace with industrial robots for manipulation operations, that will be used for educational and demonstrational purposes. Integration of sub-systems and workplace layout are worked out in this thesis. Key factor is so called virtual commissioning. Unique solution, which is contained within final thesis, combines several software tools for the design of robotic workplace. Static 3D model created in Autodesk Inventor Professional is imported into Tecnomatix Process Simulate, where dynamic 3D model is made – simulation model respectively. The control element is softPLC by Beckhoff company created in TwinCAT software. The connection of simulation model and control element is realized using standardized communication protocol OPC UA. This combination proved to be a cheap tool for virtual commissioning in practice, because only one personal computer is needed for the job.

KLÍČOVÁ SLOVA

Tecnomatix Process Simulate, digitální továrna, virtuální zprovoznění, OPC UA server, PLC Beckhoff, TwinCAT, experimentální pracoviště

KEYWORDS

Tecnomatix Process Simulate, digital factory, virtual commissioning, OPC UA server, PLC Beckhoff, TwinCAT, experimental workplace

BIBLIOGRAFICKÁ CITACE

BEDNÁŘ, M. *Návrh experimentálního robotického pracoviště pro manipulační operace*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 75 s. Vedoucí diplomové práce Ing. Aleš Pochylý.

PODĚKOVÁNÍ

Tímto děkuji panu vedoucímu diplomové práce Ing. Aleši Pochylému za jeho metodickou a odbornou pomoc bez které bych se neobešel.

ČESTNÉ PROHLÁŠENÍ

Prohlašuji, že tato práce je mým původním dílem, zpracoval jsem ji samostatně pod vedením Ing. Aleše Pochylého a s použitím literatury uvedené v seznamu.

V Brně dne 25.5.2018

.....

Bednář Martin

OBSAH

1	ÚVOD	11
2	MOTIVACE.....	13
3	PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ.....	15
4	ZADÁNÍ	17
4.1	Součást	17
4.2	Dopravník.....	18
4.3	Přípravky	20
4.4	Roboty	21
4.5	Vymezený prostor.....	22
5	KONCEPČNÍ NÁVRH PRACOVISTĚ.....	23
5.1	Varianta 1	23
5.2	Varianta 2	24
5.3	Varianta 3	25
6	VYTVOŘENÍ SIMULAČNÍHO MODELU V PS	27
6.1	Importování modelů.....	27
6.2	Manipulace s objekty	29
6.3	Koncový efektor	30
6.3.1	Návrh konstrukce.....	30
6.3.2	Implementace do SW PS.....	35
6.4	Senzory.....	36
6.5	Dopravník.....	37
6.6	Skruz	38
6.7	Příprava robotů	39
6.8	Tvorba robotických operací	40
6.9	Generování součástí.....	46
6.10	Materiálový tok	47
7	VIRTUÁLNÍ ZPROVOZNĚNÍ ROBOTICKÉ BUŇKY	49
7.1	Nastavení OPC serveru	50
7.2	Nastavení SW TwinCAT	51
7.3	Nastavení SW PS	53
7.4	Logika řízení robotického pracoviště	56
7.4.1	Robotické operace	56
7.4.2	Senzory	57
7.4.3	Signály	58
7.4.4	Procesy hlavního PLC programu.....	59
7.4.5	Vizualizace	63
8	ZHODNOCENÍ A DISKUZE	65
9	ZÁVĚR.....	67
10	SEZNAM POUŽITÝCH ZDROJŮ	69
11	SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK.....	71
11.1	Seznam tabulek.....	71
11.2	Seznam obrázků.....	71
11.3	Seznam použitých symbolů.....	72

SEZNAM PŘÍLOH	74
----------------------------	-----------

1 ÚVOD

Robotizace je nedílnou součástí průmyslů desítky let. Tento trend odstartoval počátek 20. století, kdy vznikla první montážní linka a odstartovala tak hromadnou výrobu. Daný milník je nazván Průmyslem 2.0. Vývoj šel rychle dopředu. Začaly vznikat nové požadavky. Produkce musela být přesnější (nižší tolerance), bezchybná (malé procento zmetkovitosti), flexibilní (snadné přeorientování na jiný výrobek) a rychlá (zkrácení časového cyklu). Vše bylo možné s nástupem IT technologií (30. léta 20. století), jež vdechly život pokročilé automatizaci – robotům. Ačkoliv první průmyslový robot vznikl v polovině 50. let, na propojení robotizace s IT technologií se čekalo další dvě dekády. Během této doby se snižovala cena daných periférií, zvyšoval výkon a zmenšovaly rozměry (zejména u počítačů). Počátkem 70. let nastoupil tzv. Průmysl 3.0 z důvodů výše uvedených. Navrhnout pracovní cyklus robotu pomocí PLC (Programmable Logic Controller) byla obrovská revoluce a odstartovalo období robotizace [4].

V dnešní době se společnosti setkávají se stále větším nátlakem na snižování cen, maximalizaci produktivity a zkrácení výrobního cyklu. Nicméně výroba se potýká s větší složitostí navrhovaných zařízení, jejich flexibility a variability. Tudíž je nasnadě mít si možnost už v koncepčním návrhu ověřit takt celé linky a její prostorové nároky. Průmysl 4.0 s sebou přináší východisko v podobě tzv. virtuálního zprovoznění (virtual commissioning). Pomocí 3D modelů komponent linky je možné otestovat navržené pracoviště včetně logiky řídicích prvků. Díky neustálému vývoji v oblasti softwarových nástrojů se nabízí možnost vytvořit komplexní virtuální prostor, jenž „identicky“ napodobuje prostor fyzický [12].

Právě tato práce se zaměřuje na návrh robotického pracoviště ve virtuálním prostředí, kdy pomocí sofistikovaného softwaru Tecnomatix Process Simulate (dále jen „PS“) dokážeme ověřit, zdali koncepční layout vyhovuje všem požadavkům. Následně PS umožňuje optimalizovat jednotlivé operace pro roboty, aby se dosáhlo, co možná nejlepšího výsledku. Klíčovou fází této diplomové práce bylo dále navržené pracoviště v PS tzv. virtuálně zprovoznit, resp. propojit s nadřazeným PLC systémem a otestovat tak různé funkční scénáře na daném pracovišti ještě před fyzickou realizací, což významně usnadní a zkrátí uvádění linky do provozu.

2 MOTIVACE

V předešlých letech se diplomové práce se zaměřením na konstrukci robotického pracoviště opírali pouze o statický 3D model vytvořený v CAD softwaru, díky čemuž bylo možné ověřit celkový rozměr pracoviště a rámcové dosahy jednotlivých robotů. Nicméně časy jednotlivých operací a funkčnost navržené logiky celého pracoviště nebylo možné otestovat. Což je nedostačující řešení pro tak komplexní úlohu s ohledem na možný výskyt velkého množství chyb.

Diplomová práce reaguje na požadavek trhu. Rostoucí globální konkurence zesiluje tlak v oblasti automatizace. Vznikají požadavky na složitější a individuální automatizační řešení v co nejkratším možném čase. Uvedení automatizované linky do provozu je jedna z časově a kapitálově náročných etap. Pokud se při ožívování pracoviště nalezne chyba či je nutná úprava a značně se tím zpozdí předání zákazníkovi. Aby bylo možné dobu zprovoznění zkrátit a zároveň eliminovat chyby řešení, je vhodné mít k dispozici integrované a flexibilní řešení [3]. K tomuto účelu slouží tzv. virtuální zprovoznění, jež prakticky „vybuduje“ celou automatizovanou linku v kybernetickém prostředí. Následné testování různých scénářů jednoduše ověří nejlepší možné řešení. Virtuální zprovoznění se stále řadí do fáze návrhu, a proto nově požadované úpravy na komponentách jsou časově nenáročné v porovnání s úpravou již vyrobených komponentů.

Unikátní řešení, jež je obsaženo v diplomové práci, kombinuje při návrhu robotického pracoviště více softwarových nástrojů. Statický 3D model vytvořený v Autodesk Inventor Professional se importuje do PS, kde se tak vytváří 3D model dynamický. Následné propojení PS s PLC řízením společnosti Beckhoff pomocí OPC serveru umožňuje detailní pohled na problematiku daného projektu. Ukázalo se, že tato kombinace (Obr. 1) může pro praxi představovat levný nástroj pro virtuální zprovoznění, jelikož je zapotřebí pouze jednoho osobního počítače.



Obr. 1) Schématické propojení jednotlivých softwarových nástrojů

3 PŘEHLED SOUČASNÉHO STAVU POZNÁNÍ

Majoritní část práce je tvořena v softwaru určeném pro simulaci a navrhování robotických linek. Jedná se o program Siemens Tecnomatix Process Simulate (dostupný na FSI VUT v Brně, včetně RCS modulu pro roboty KUKA) a jeho význam je daný především díky značnému rozšíření v oblasti automotive. Používá se jako standard v rámci koncernu VW/Škoda, BMW. Využití nalézá i u předních integrátorů robotických pracovišť v ČR (Blumenbecker Prag s.r.o, apod.).

Program PS (konkrétně Robotics) umožňuje plánovat pro roboty operace jako: manipulace, balení, montáž, leštění, řezání laserem a vodním paprskem, svařování, lakování a další. Samozřejmostí je i offline programování s převedením do nativního jazyka robota. Pro snadnější plánování robotických operací slouží funkce *Automatic Path Planner* a detekce kolizí během simulace. Program nabízí podporu pro týmovou spolupráci na tentýž projektu se správou dat pomocí Oracle databáze. Existuje přímá návaznost na aplikace:

- Process Designer — ověřuje navržený layout výrobní linky a naplánované časy dílčích operací pomocí dynamické simulace. Umožňuje vyhodnotit výrobní alternativy, koordinovat prostředky, plánovat více variant, zavádět změny a odhadovat náklady.
- Plant Simulation — umožňuje dynamickou simulaci diskrétních událostí na úrovni celé výrobní haly. Lze vytvářet digitální modely výrobních a logistických systémů a následně zkoumat charakteristiky systémů a optimalizovat jejich výkonnost. Za zmínku stojí možnost denní operativní plánování výroby s cílem maximálního využití zdrojů při změnách vstupů [19, 11].

V současné době nabízí několik dalších společností software pro digitální vývoj průmyslu, které mohou více či méně konkurovat systému PS. Správná volba není snadná a často vyžaduje určitou odbornost a praxi v této oblasti. Nejznámější alternativy jsou: ABB RobotStudio a DELMIA Robotics.

❖ ABB RobotStudio

Program (Obr. 2) sloužící pro offline programování zaměřené na roboty firmy ABB. Veškeré simulace jsou založeny na virtuálním řídicím systému firmy ABB – přesné kopii opravdového softwaru, jenž řídí reálné roboty ve výrobě [8]. Proto umožňuje velmi realistické simulace s využitím robotických programů a konfiguračních souborů, které jsou identické jako programy a soubory přímo ve výrobě. Jako přednost se jeví import CAD souborů v různých formátech. Jedná se např. o: IGES, STEP, VRML, VDAFS, ACIS a CATIA. Dále nabízí pokročilé funkce pro usnadnění tvorby jednotlivých cest operací pro roboty. Funkce *AutoPath* pomáhá generovat body podle kontury objektu. A pokud se během sledování kontury dostane robot do singulární polohy, pomocí automatické detekce funkce *Path Optimization* se lze tomuto nechtěnému stavu vyhnout. Jakmile je simulace spuštěna, lze v aplikaci přímo zasahovat do programového kódu řídicího systému (funkce *RAPID*) a optimalizovat tak svou činnost [7].

❖ DELMIA Robotics

Produkt spadá do platformy 3DEXPERIENCE, jež slouží ke kompletnímu řízení podniku. Nicméně klíčovým produktem pro robotické simulace je právě DELMIA Robotics, viz Obr. 2. Software obsahuje komplexní řešení pro offline programování. Nabízí několik klíčových modulů:

- DELMIA Robotics Offline Programming

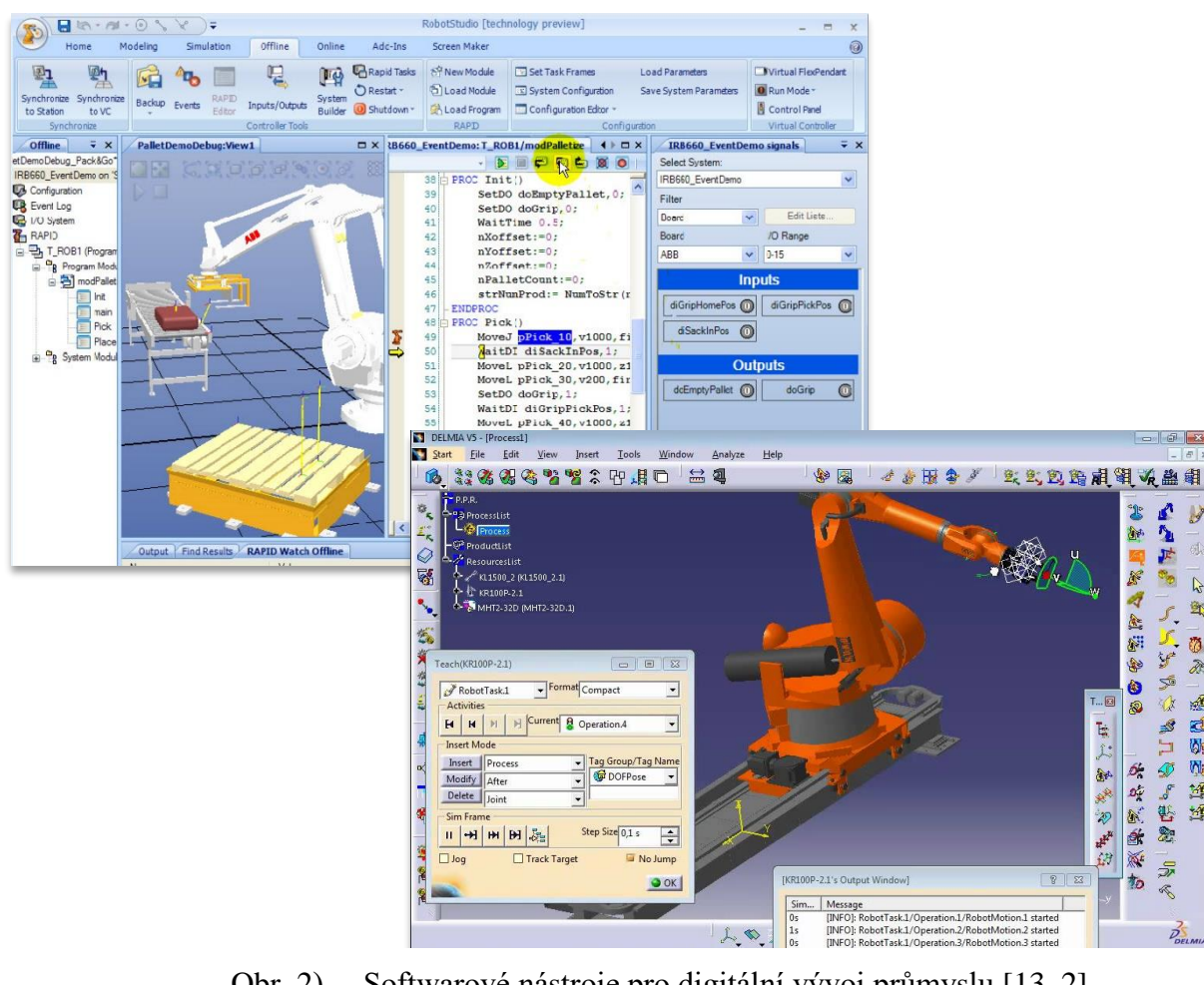
Modul umožňující import a export robotických programů. Operacím vyžadující vysokou přesnost trajektorií nabízí postprocesor pokročilé kalibrační schopnosti zajišťující přesnost přeloženého programu [5].

- DELMIA Robotics Task Definition

Modul poskytuje 3D prostředí, kde si mohou uživatelé vytvářet, programovat, simulovat a ověřovat robotické pracoviště. Simulanti zde mohou umístit vlastní periferie a roboty a propojit navzájem jejich řídicí systémy. Modul obsahuje rozsáhlou knihovnu robotů a kontrolérů od všech významných průmyslových výrobců [5].

- DELMIA Robotics Virtual Commissioning

Důležitý modul určený pro tzv. virtuální zprovoznění. Pomocí implementovaného standardu Realistic Robot Simulation II (RRS-II) se provádí validace robotických programů s téměř 100% přesností ve srovnání s fyzickým robotem [6].



Obr. 2) Softwarové nástroje pro digitální vývoj průmyslu [13, 2]
vlevo nahoře — ABB RobotStudio; vpravo dole — DELMIA Robotics

4 ZADÁNÍ

Cílem diplomové práce je návrh experimentální robotické buňky pro manipulační operace. Hlavní funkcí pracoviště bude edukace studentů nejen navazujícího magisterského oboru Výrobní stroje, systémy a roboty, ale i ostatních oborů, které mají ve studijním plánu předmět týkající se výuky robotů. Pracoviště má nabízet částečný pohled do automatizované výroby a přiblížit aplikaci šestiosých robotů s dílčími systémy v praxi.

Návrh reálného pracoviště pro některou zvolenou manipulační operaci by bylo kontraproduktivní. Není cílem prezentovat robotickou buňku (resp. její část), jež lze spatřit ve výrobních prostředích. Zadání projektu si klade za důraz, aby výsledné řešení, jež je výstupem diplomové práce, bylo co nejvíce komplexní. Myšleno tak, aby se objevilo co možná největší množství různých interakcí robotů s periferiemi, které se často objevují v manipulačních operacích.

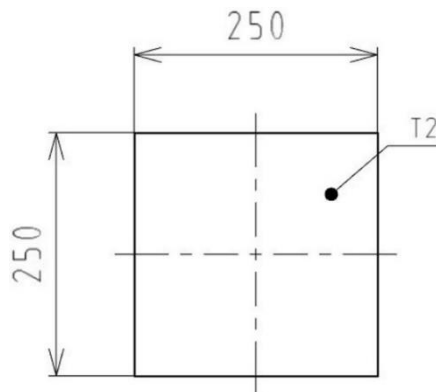
Prvním cílem je vytvoření koncepčního návrhu dle požadavků výše uvedených za předpokladu použití dostupných komponent (roboty, pedestaly, dopravník, přípravky), viz následující podkapitoly. Návrh musí rozměrově odpovídat vymezenému prostoru, viz kapitola 4.5. Následně se pomocí 3D objektů vytvoří model pracoviště v prostředí PS. Klíčovou fází je tzv. virtuální zprovoznění. Model pracoviště se obohatí o robotické operace a senzory. Jednotlivé periferie se pomocí funkcí v PS změň na interaktivní modely. Vytvoří se jednotlivé signály, jež mohou komunikovat s OPC serverem. Na druhé straně PLC Beckhoff vytvoří tytéž signály pro získávání informací a posílání instrukcí pomocí nadefinovaného řídicího programu. To vše pomocí jednoho osobního počítače, na kterém se budou nacházet veškeré softwarové nástroje relevantní pro tzv. virtuální zprovoznění.

4.1 Součást

Projekt nemá v úmyslu se ubírat směrem k manipulaci tvarově složitých součástí, pro nichž je zapotřebí konstrukce specifického efektoru, jelikož se jedná o pilotní verzi pracoviště. Proto je pro lepší demonstraci zadána tvarově jednoduchá součást – plech čtvercového profilu o rozměrech 250 x 250 x 2 mm (Obr. 3). Tvar součásti zastupuje polotovary určené k dalšímu technologickému zpracování například: tažení, ohýbání, děrování, vysekávání, přesné stříhání apod.

Aby byla zastoupena částečná variabilita pracoviště, projekt zahrnuje schopnost manipulace veškerých součástí (za předpokladu drobných změn) splňující podmínky:

- ✓ šířka od 150 mm do 290 mm,
- ✓ délka od 150 mm do 290 mm,
- ✓ tloušťka od 1 mm,
- ✓ hmotnost do 2 kg,
- ✓ povrch nečlenitý a hladký.



Obr. 3) Rozměry manipulované součásti

4.2 Dopravník

Dopravníky slouží k přesunu dílce z jednoho místa na druhé. Implementují se do robotické buňky pro: doručení součástí do vstupního místa, přesun z výstupního místa na jiné pracoviště nebo posun dílce uvnitř buňky. Nespornou výhodou je, že operátor se nemusí pohybovat v potenciálně nebezpečné zóně, kde se nachází pohybující se roboty. V případě, že by funkci dopravníků nahradil člověk, musela by robotická ruka přerušit svou práci a čekat, až operátor vystoupí z pracovní oblasti. To by zapříčinilo prodloužení celkového času a takového jednání by bylo velice neefektivní z hlediska automatizované linky jako takové.

Pro zadaný projekt zákazník zvolil dodavatele ALUTEC K & K, s.r.o. Jedná se o známou českou firmu fungující od roku 1996. Tato volba přináší řadu výhod. Dodací časy jsou relativně krátké, tudíž lze řešit případné nesrovnalosti operativně přímo na místě. Zákazník je dlouhodobým odběratelem různého sortimentu od zmíněné firmy.

Projekt musí respektovat již dostupný pásový dopravník DP50 2000x350 (Obr. 4) s asynchronním motorem BN 63B 4 od firmy Bonfiglioli a šnekovou převodovkou VF30 A P63 B14. Řízení pohonu obstarává frekvenční měnič Commander SK od firmy Emerson. Jelikož je pásový dopravník pro konkrétní aplikaci značně předimenzován, pro správný návrh pracoviště stačí znát pouze klíčové parametry, jež prezentuje tabulka níže.

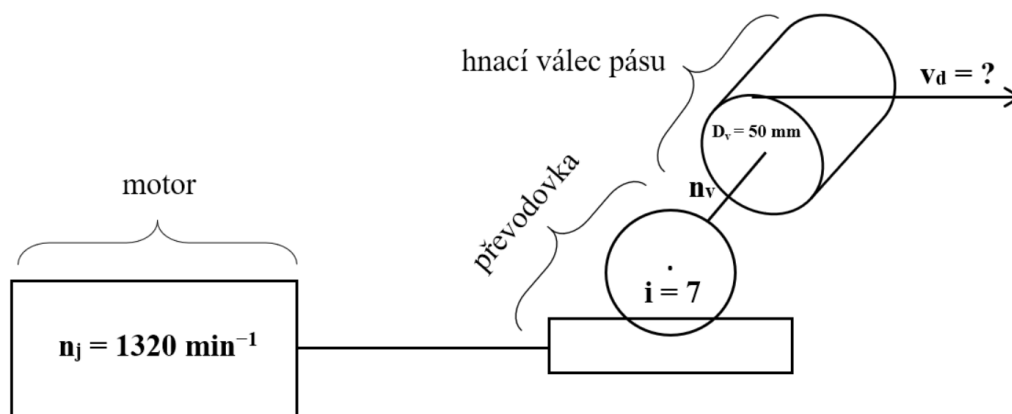
Tab 1) Základní parametry dopravníku, pohonu a převodovky

Dopravník DP50 2000x350			
název	symbol	hodnota	jednotka
šířka	$L_{dš}$	350	[mm]
délka	L_{dd}	2000	[mm]
výška	L_{dv}	800	[mm]
průměr hnacího válce	D_v	50	[mm]
Pohon BN 63B 4			
název	symbol	hodnota	jednotka
jmenovitý točivý moment	M_j	1,3	[Nm]
jmenovité otáčky	n_j	1320	[min ⁻¹]
Převodovka VF30 A P63 B14			
název	symbol	hodnota	jednotka
převodový poměr	i	7	[–]



Obr. 4) Dopravník DP50 2000x350

Předpokládá se, že celkové zatížení dopravníku ve výsledném řešení bude výrazně nižší než zatížení dovolené. Proto jsou na dopravník kladeny pouze rychlostní požadavky. Základní parametry v Tabulce 1 postačí k výpočtu maximální rychlosti pásu „ v_d “ pomocí vztahu (4.1) pro obvodovou rychlost dle schématu znázorněném na Obrázku 5.



Obr. 5) Kinematické schéma pohonu dopravníku

$$v_d = \omega \cdot r \quad (4.1)$$

$$v_d = 2\pi \frac{n_v}{60} \cdot \frac{D_v}{2} \quad (4.2)$$

$$v_d = 2\pi \frac{n_j}{60} \cdot \frac{1}{i} \cdot \frac{D_v}{2} \quad (4.3)$$

$$v_d = 2\pi \frac{1320}{60} \cdot \frac{1}{7} \cdot \frac{50}{2} = 493,68 \frac{mm}{sec}$$

kde n_v — jmenovité otáčky hnacího válce dopravníku [min^{-1}].

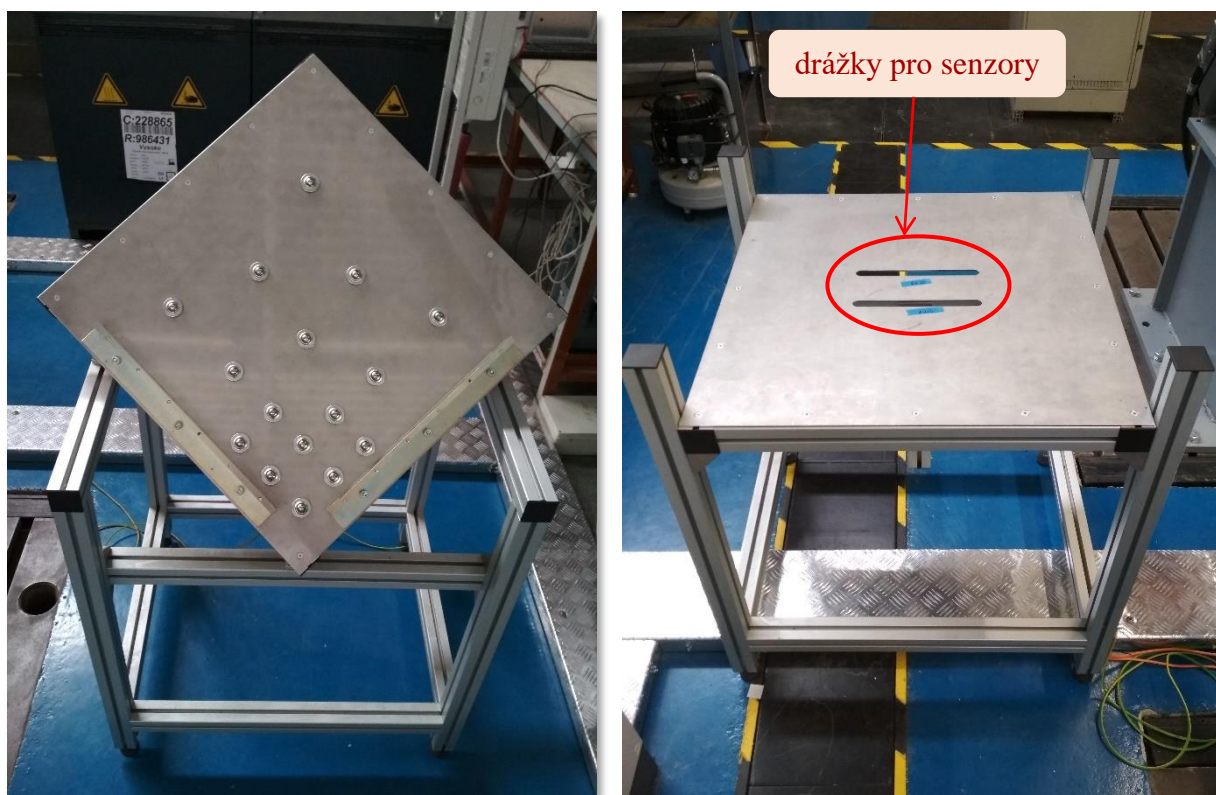
Po zaokrouhlení vychází, že maximální požadovaná rychlost dopravníku by neměla přesáhnout $494 \text{ mm} \cdot \text{sec}^{-1}$.

4.3 Přípravky

K zadanému projektu jsou již dostupné dva přípravky. Jejich nosné soustavy jsou vyhotoveny z modulárních dílů firmy ALUTEC K & K, s.r.o. a zbytek dílů vyrobeny na zakázku.

První přípravek – stůl (Obr. 6) slouží k definování vstupní a výstupní pozice robotického pracoviště. Předpokládá se, že bude vytvořen stoh ze součástí, jež se bude postupně odebírat. Obdobný předpoklad je vznesen pro paletizaci na výstupní pozici. Na odkládací ploše jsou vytvořeny dvě drážky sloužící k umístění senzorů pro obě místa. Odkládací plocha o rozměrech 600 x 600 mm je připravena pro součásti o maximálních rozměrech 290 x 290 mm, jelikož je nutné ponechat prostor pro mezeru mezi pomyslnými odkládacími místy.

Předpokládá se, že vstupní stoh je mírně nepřesný vlivem zakládání součástí do robotického pracoviště operátorem. Proto je vhodné umístit do pracoviště mezičlánek. Přípravek zajišťující vycentrování součásti pro uchopení efektem, tj. skluz. Jakmile robot položí součást do prostoru skluzu, díl se vlivem tíhové síly přesune do předem známé pozice. O hladký přesun se stará 16 valivých hnízd s postupně zvětšující se roztečí, viz Obr. 6. Zábrany umístěné na spodních stranách čtvercové skluzové plochy vymezují centrovací pozici. Úhel spádu skluzové plochy lze přestavit, přičemž se předpokládá nejvhodnější spádový úhel 45 °. Přípravek vznáší předpoklad, že součást bude mít nejmenší půdorysnou plochu o rozměrech 150 x 150 mm.



Obr. 6) Dostupné přípravky
vlevo — skluz; vpravo — stůl

4.4 Roboty

Roboty jsou bezesporu klíčovou periferií v automatizovaném pracovišti. Pro správnou volbu robotu, je vhodné si nejdříve určit, o jaký typ operace se jedná. Každá operace má své specifikum.

- Obráběcí operace vyžaduje vysokou přesnost, tuhost a opakovatelnost dráhy pohybu.
- Svařování klade důraz na kolmost nástroje vůči aktuálnímu bodu na povrchu svařovaného objektu, tj. normálový vektor musí být totožný s osou (nejčastěji osa z) nástrojového souřadného systému.
- Manipulační operace požaduje vyšší dynamiku, tj. rychlý pohyb i za předpokladu těžšího břemene.

Pro roboty existují tři typy kinematických řetězců: sériový, paralelní a hybridní. Hybridní kinematika se obvykle skládá od základny z paralelní kinematiky (paralelogram) a rameno u efektoru je tvořeno sériovým otevřeným kinematickým řetězcem. Hybridní kinematická struktura se používá ze všech nejméně často a její využití je u speciálních operací nebo běžných operací, kde je kladen vysoký nárok na určitou vlastnost robota. Mezi další důležité aspekty se řadí nosnost, počet stupňů volnosti, stacionárnost a pracovní prostor robota.

V zadaném projektu figurují dva šestiosé průmyslové roboty. Jedná se o roboty KR 5 ARC a KR 16-2 od firmy Kuka umístěné na stacionárním pedestalu, viz Obrázek 7. Jelikož jsou typy předem zadané, resp. dostupné, je potřeba v zadaném projektu počítat s jejich reálnými vlastnostmi. Tabulka níže prezentuje základní parametry, jež je nutné dodržet pro úspěšné nasazení robotů do konkrétního řešení.

Tab 2) Základní parametry robotů [14, 15]

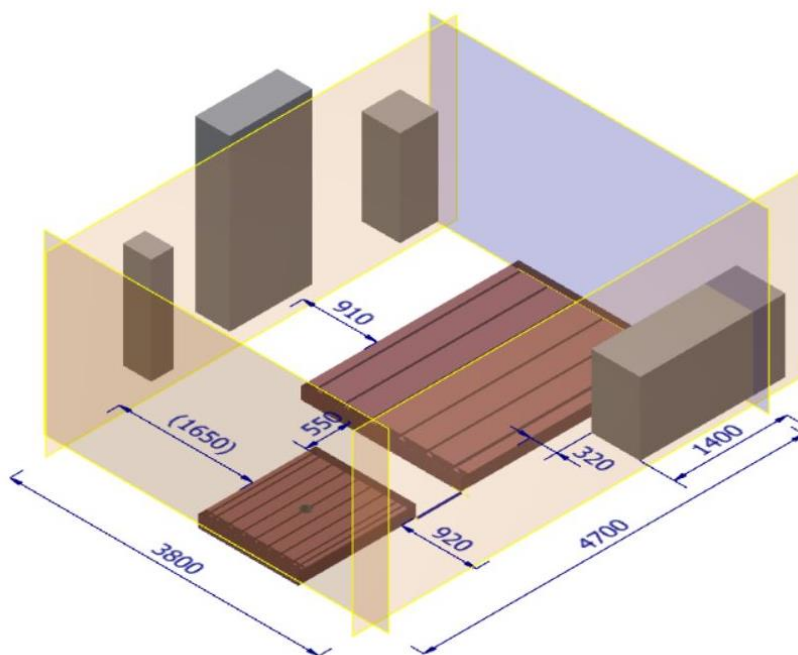
KR 5 ARC			KR 16-2		
5	[kg]	jmenovitá nosnost	16	[kg]	
1530	[mm]	maximální dosah	1611	[mm]	
127	[kg]	celková hmotnost	235	[kg]	



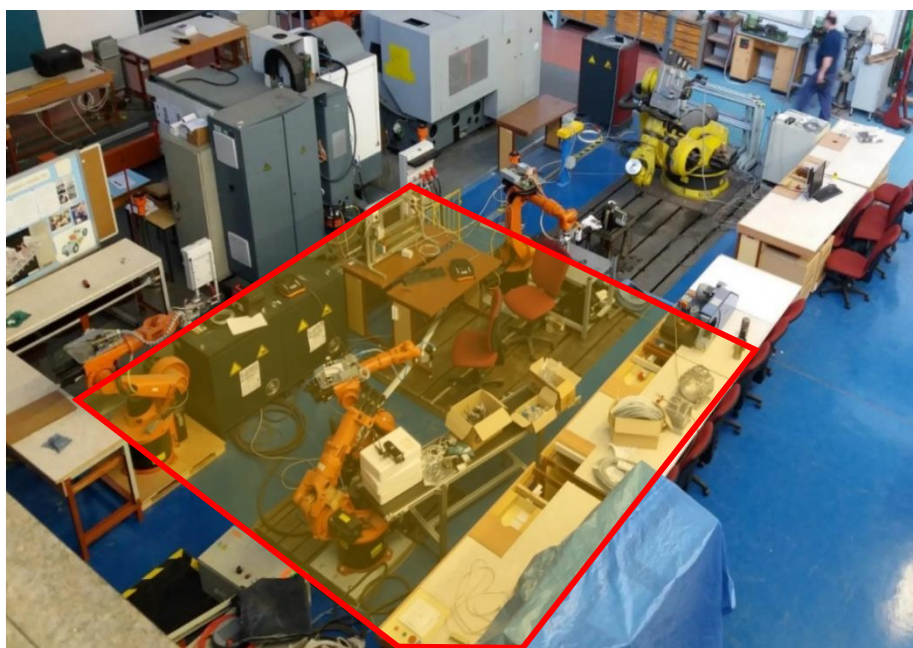
Obr. 7) Modely robotů KUKA
 vlevo — KR 5 ARC; vpravo — KR 16-2

4.5 Vymezený prostor

Pro robotickou buňku je vymezen prostor v areálu VUT FSI v zóně C1, viz Obrázek 8. Jedná se o halu, kde se mimo jiné nachází frézovací centra a další roboty. Hala nabízí multifunkční využití s dílenskými nástroji. Prostor pro experimentální pracoviště je vymezen dle upínacích desek zabudovaných v podlaze o rozměrech 2700 x 1980 mm a druhá o rozměrech 1460 x 1240 mm. Na Obrázku 9 lze vidět výsledný pracovní prostor pro návrh layoutu (ohrazen rovinami) o rozměrech 3800 x 4700 mm. Do prostoru částečně zasahuje pracovní stůl a s jeho přítomností je nutno počítat.



Obr. 9) Základní rozměry pracovního prostoru



Obr. 8) Nevyklizený pracovní prostor v hale

5 KONCEPČNÍ NÁVRH PRACOVISTĚ

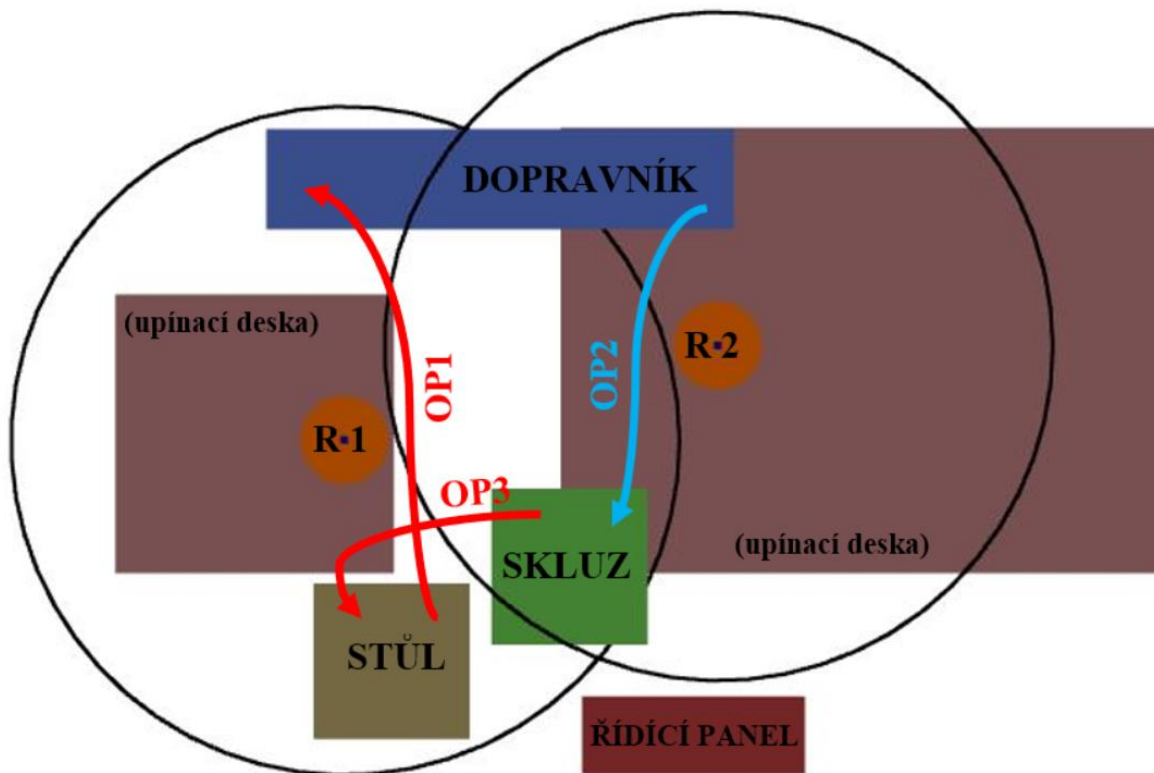
Prvním krokem k vytvoření finální verze pracoviště je vytvoření konceptu. Výstupem je layout, v němž se nachází hrubě načrtnuté klíčové periferie s vhodným umístěním. Prvotní návrh je tvořen zejména robotikem, resp. simulantem. V dalších fázích návrhu se postupně zapojuje konstruktér a pracuje se na projektu simultánně.

Jelikož jsou předem známe téměř veškeré klíčové periferie, není problém sestavit několik koncepčních rozvržení. Finální verze layoutu se nicméně může od konceptu mírně lišit, pokud se díky simulacím zjistí, že je nutné některou z periférií posunout. Proto z důvodu zachování částečné flexibility rozvržení volím do koncepčního layoutu pracovní obálku (kružnici) o poloměru 1500 mm pro oba roboty i za předpokladu, že disponují maximálním dosahem 1530 mm a 1611 mm, viz Tabulka 2.

V následujících podkapitolách jsou uvedeny tři možné koncepce uspořádání pracoviště splňující zadání. Nicméně diplomová práce rozpracuje pouze variantou č. 3 z důvodů uvedených přímo u jednotlivých variant.

5.1 Varianta 1

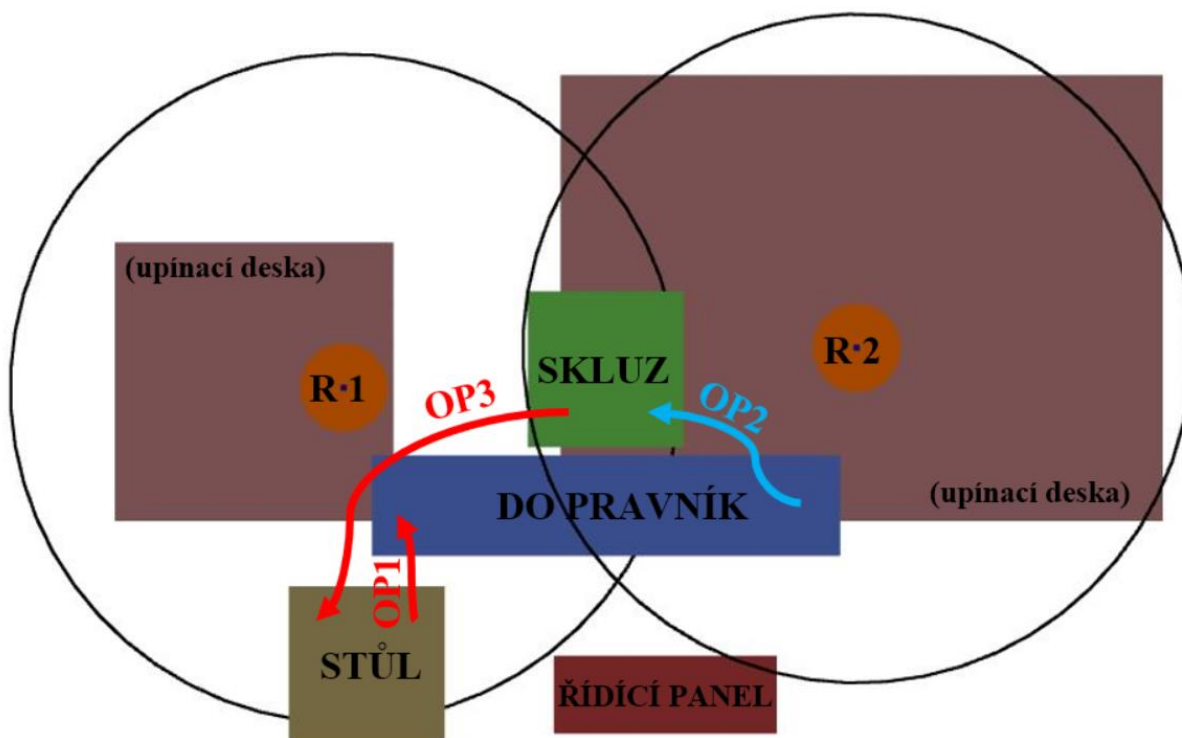
Robot R1 obstarává operaci OP1 a OP3 a robot R2 operaci OP2. Délky jednotlivých operací nejsou dlouhé. Nicméně roboty jsou relativně blízko u sebe. Pracovní obálka prvního robota by neměla zasahovat do těla robota druhého, což se v této variantě stane, jakmile se rameno robota prodlouží o výšku efektoru. Veškeré operace procházejí společným pracovním prostorem, což je nežádoucího vzhledem k hrozcím kolizím.



Obr. 10) Varianta 1 – koncepční layout

5.2 Varianta 2

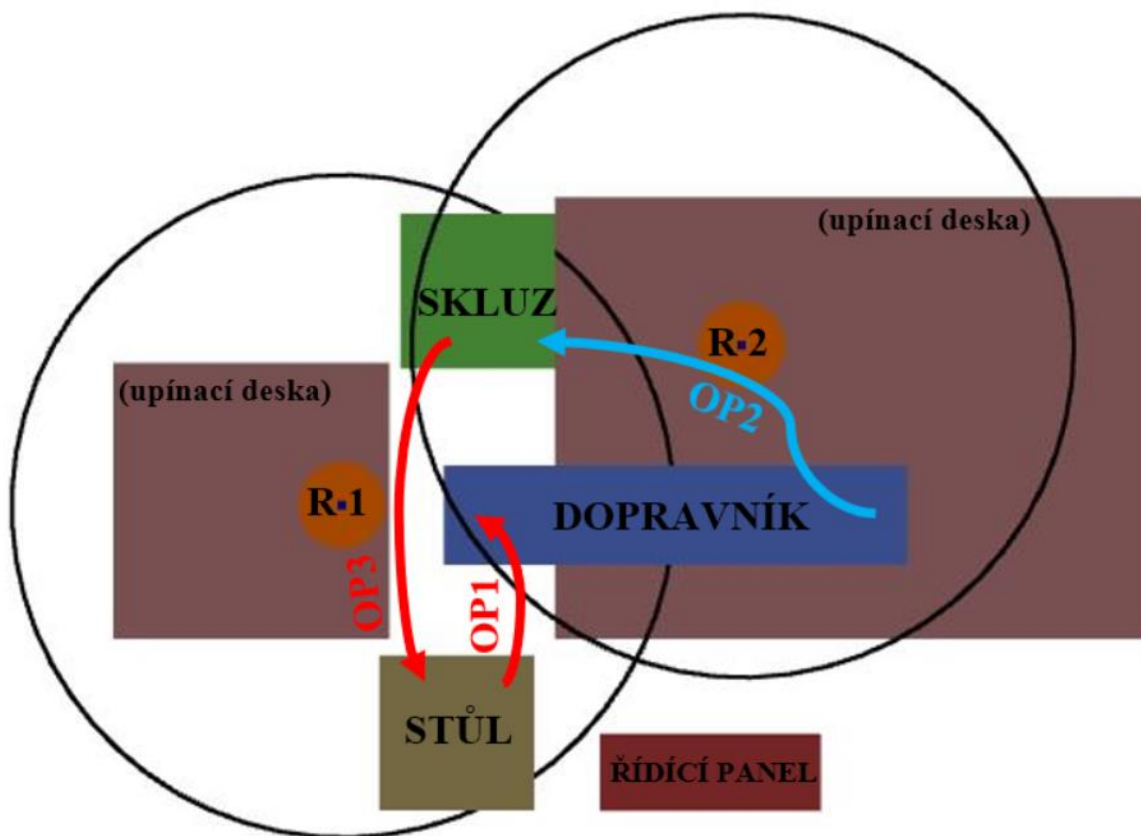
Další varianta se jeví jako lepší řešení. V ve sdílené pracovní zóně robotů se nachází pouze skluz a střed dopravníku. A jelikož se roboty na přípravku – skluz střídají, není zde vysoké riziko kolize. Nicméně dráhy jednotlivých operací jsou příliš krátké a v experimentálním manipulačním pracovišti by zřejmě plně nevynikla činnost robotů.



Obr. 11) Varianta 2 – koncepční layout

5.3 Varianta 3

Tato varianta je velmi podobná druhé variantě. Skluz je přesunut do nejvzdálenějšího místa, kde se stále protínají pracovní obálky obou robotů. Délky operací OP2 a OP3 jsou prodlouženy. Celková demonstrační ukázka manipulace by měla vypadat nejlépe ze všech tří variant, což je i jedním z důvodů, že tato varianta bude dále rozpracována.



Obr. 12) Varianta 3 – koncepční layout

6 VYTVOŘENÍ SIMULAČNÍHO MODELU V PS

Kapitola podrobně popisuje postup, jak lze zadaný projekt, resp. jeho simulační model vypracovat v PS. Podle následujících kroků lze postupovat i při vypracování jiné úlohy. Je však potřeba si uvědomit, že takto komplexní software nelze zcela pochopit pouze z jednoho návodu. Doporučuje se absolvovat řádné školení. PS umožňuje vytvořit dva typy simulací níže uvedených.

❖ Sekvenční (*Standard Mode*)

V konvenční simulaci založené na čase se určuje předdefinovaná sekvence operací pro simulaci procesu [10]. Základním pilířem je Ganttův diagram, resp. *Sequence Editor*. Vodorovná osa značí časovou linii a osa horizontální jednotlivé operace v procesu. Použití sekvenční simulace je vhodné pouze pro procesy, kde je známý sled operací a není potřeba v průběhu simulace logického rozhodování.

❖ Event-based (*Line Simulation Mode*)

Umožňuje simulovat výrobní stanice, kde musí v plné synchronizaci fungovat řada robotů, výrobních zdrojů a řídicích zařízení. Průběh simulace se určí podle logiky řídicího členu. V reálném čase se reaguje na změny hodnot signálů a díky tomu může být každá simulace stejného procesu jedinečná [10].

Následující návod prezentuje vytvoření simulačního modelu s přípravou pro tzv. virtuální zprovoznění. Tudíž konvenční způsob je nedostačující a použije se vyšší úroveň simulace, tj. *Event-based simulation*.

6.1 Importování modelů

Konstrukčně navržené periferie se do prostředí PS primárně importují (nejčastěji ve formátu .JT) z jiných návrhových 3D modelářů. Program PS nabízí sice jednoduché nástroje pro vymodelování 3D objektů, nicméně nenabízí rozšířené funkce jako plnohodnotný CAD software.

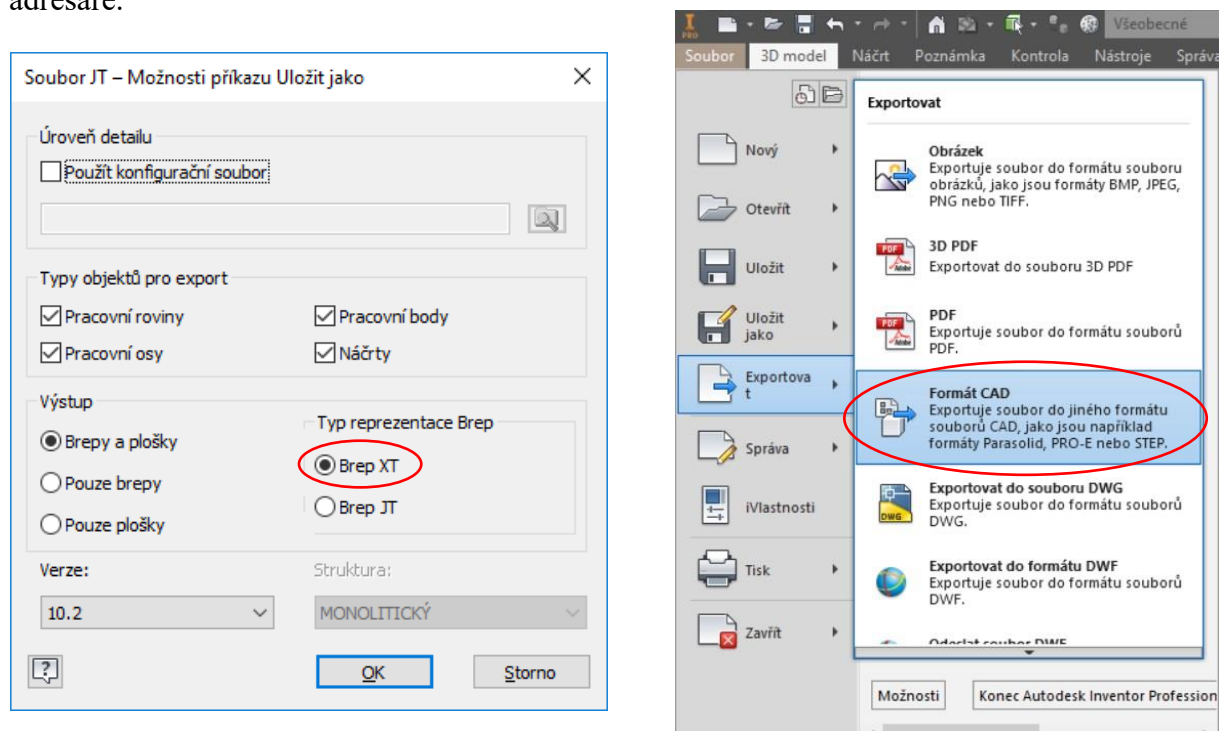
Při vytváření 3D modelů, jež reprezentují reálné komponenty v pracovišti, je nutné dbát na přesnost. Preciznost a rozměrová přesnost se promítne ve výsledném ožívování pracoviště. Pokud jsou komponenty kupované, je vhodné CAD soubory stáhnout přímo od výrobce. Jsou často do jisté míry exaktní vůči skutečnosti. To se nicméně může jevit i jako nevýhoda. V některých případech, kdy je pracoviště rozsáhlé, se lze dostat do bodu, kdy dostupný výpočetní výkon nemusí dostávat nebo se značně zpomalí práce či generování výsledné simulace.

Je-li nutné z důvodu nedostupnosti model vytvořit, volí se CAD software, jenž umožňuje export ve formátu .JT. Výpis níže prezentuje některé softwary podporující zmiňovaný formát:

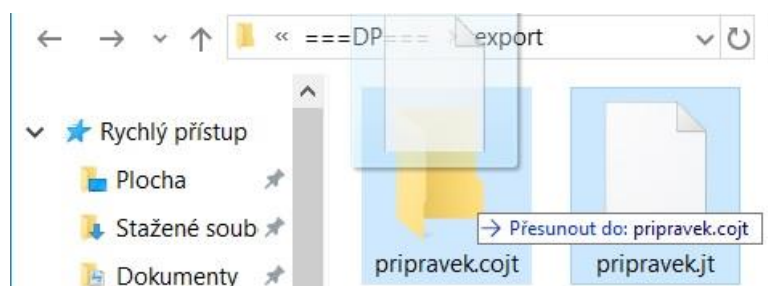
- | | |
|----------------------------|-----------------------|
| ✓ Creo Parametric, | ✓ Siemens Solid Edge, |
| ✓ Dassault Systèmes CATIA, | ✓ Autodesk Inventor. |
| ✓ Siemens NX, | |

Pro vytvoření 3D modelů volím software od společnosti Autodesk s názvem Inventor Professional 2018, jelikož jsem s ním nejvíce obeznámen. Získání CAD souboru s příponou .JT je následující. Před exportem doporučuji navíc součást předem obarvit jednolitou barvou pro snazší pozdější orientaci v sestavě. Vytvořená součást nebo sestava se vyexportuje možností

Formát CAD, viz Obrázek 13. Dále se změní přípona v nabídce *Uložit jako typ*: na *Soubory JT* (*.jt). Před samotným uložením doporučuji zkontrolovat v nabídce *Možnosti*, zdali se soubor bude ukládat typem prezentace *Brep XT*, viz Obrázek 13. Typ prezentace XT je novější než JT a umožňuje věrně převádět 3D objekty do jiných CAx softwarů třetích stran. Zajišťuje lepší kompatibilitu a opravuje některé geometrické chyby [9]. Zbývá pouze vytvořit stejnojmennou složku a do názvu připsat řetězec znaků .cojt a následně soubor do složky překopírovat dle Obrázku 14. Vytvořené .cojt „soubory“ (reálně se jedná o složky) se přesunou do libovolného adresáře.

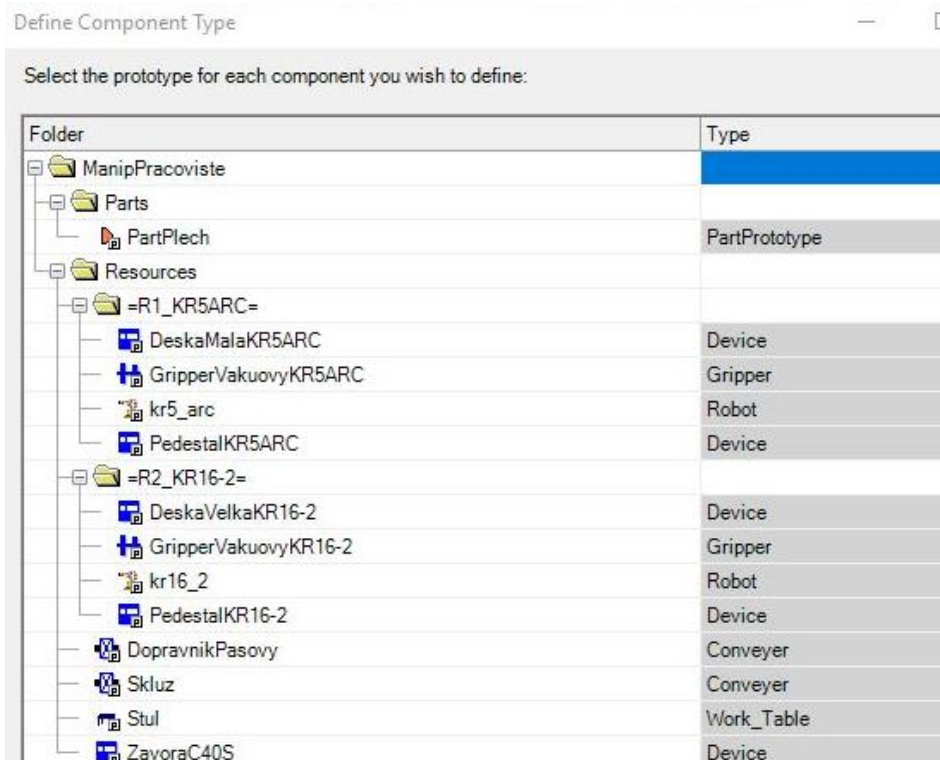


Obr. 13) Nastavení souboru pro správný export do *.JT
vlevo — možnosti .JT exportu; vpravo — cesta k exportování



Obr. 14) Vytvoření .cojt „souboru“

Dalším krokem je definování typu komponenty. Každý objekt, jenž bude používán v PS, musí mít stanovený, o jaký typ se jedná. K tomu slouží funkce *Modeling* → *Components* → *Define Component Type*, kde se následně zvolí výše zmiňovaný adresář s .cojt „soubory“. Pokud se typ komponenty v seznamu nenalézá, doporučuji zvolit *Device*, jelikož se tato možnost jeví jako univerzální. V případě zadaného projektu se komponenty klasifikují dle Obrázku 15.

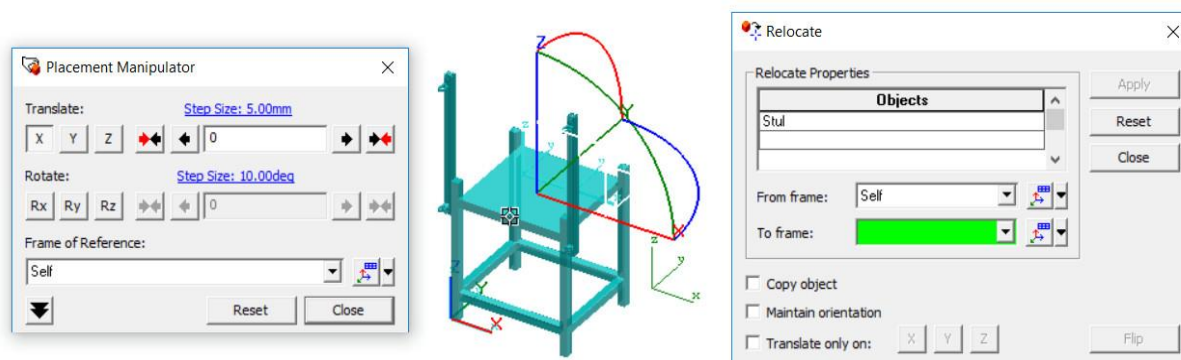


Obr. 15) Definování typu komponent v zadaném projektu

Při vkládání jednotlivých objektů (*resources*) se pro větší přehlednost používá *Modeling* → *Components* → *Compound Resource*. Funkce pomáhá větvení a logickému skupinovému uspořádání *resources* v *Object tree*. A pro samotné vložení jednotlivých .cojt „souborů“ je určena funkce *Modeling* → *Components* → *Insert Component*.

6.2 Manipulace s objekty

Jelikož se veškeré importované komponenty zjeví v počátku výchozího souřadného systému, je nutné objekty přesunout dle určitého rozvržení. Pro jednoduchý přesun v kartézském souřadném systému slouží funkce *Placement Manipulator*. Ke specifitějšímu přesunu na předem určené místo (pomocí *frame*) slouží funkce *Relocate*. Obě možnosti se zjeví po pravém kliknutí na požadovaný objekt v grafickém okně.



Obr. 16) Funkce pro přesun objektů
vlevo — *Placement Manipulator*; střed — objekt pro přesunutí; vpravo — *Relocate*

Nejen k manipulaci s objekty se používá výše zmíněný *frame*. Jedná se klíčovou funkcí programu PS. V *Modeling* → *Layout* → *Create Frame* se vytvoří lokální souřadný systém pomocí čtyř různých metod. *Frame* je vhodný si předem umístit na strategická místa. Jedná se například o:

- ✓ roh modelu — vztažný bod dle kterého se odměří pozice součásti ve vytvořeném layoutu. Doporučuji levý spodní roh z předního pohledu 3D layoutu z důvodu snadnějšího nadefinování nového *frame* metodou *Frame by 3 points* a následného přemístění objektu pomocí funkce *Relocate*.
- ✓ Střed odkladové zóny — místo, kde se nachází střed součásti nebo jiné logické místo.
- ✓ Střed horní desky pedestalu — vhodný, jelikož robot vlastní výchozí *frame* s názvem *self*, jenž se nachází ve středu jeho podstavu.
- ✓ Střed součásti — *frame* na rubové straně pro umístění např. na stůl a *frame* na lícové straně pro uchopení efektořem.
- ✓ Vstupní a výstupní místo nacházející se na konkrétní periférii např. na dopravníku nebo skluzu.

Výše zmíněná strategická místa se týkají konkrétního objektu. Tudíž je vznesen požadavek, aby *frame* byl svázaný s daným objektem. Po kliknutí na konkrétní součást v grafickém okně ji lze odemknout pro modelování pomocí *Modeling* → *Scope* → *Set Modeling Scope*. Nyní je možné vytvářet *frame*, jenž bude součástí modelu. Aby byl daný *frame* vždy viditelný a šel použit na další činnosti během tvorby simulace, je nutné ho zviditelnit pomocí *Modeling* → *Entity Level* → *Set Object to be Preserved*. Po dokončení nezapomenout ukončit úpravy modelu pomocí *Modeling* → *Scope* → *End Modeling*.

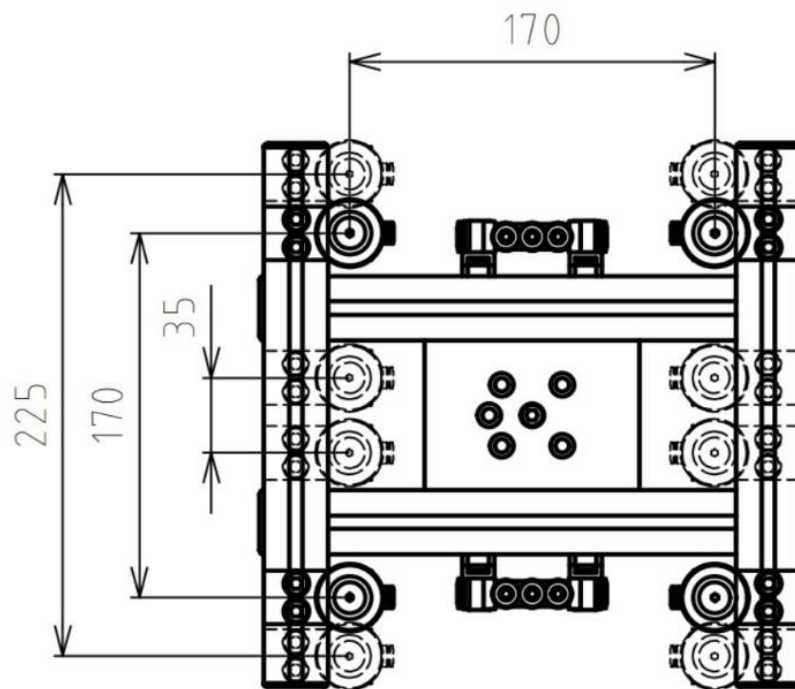
6.3 Koncový efektor

Jelikož není pro zadaný projekt žádný efektor k dispozici, volím vlastní konstrukci.

6.3.1 Návrh konstrukce

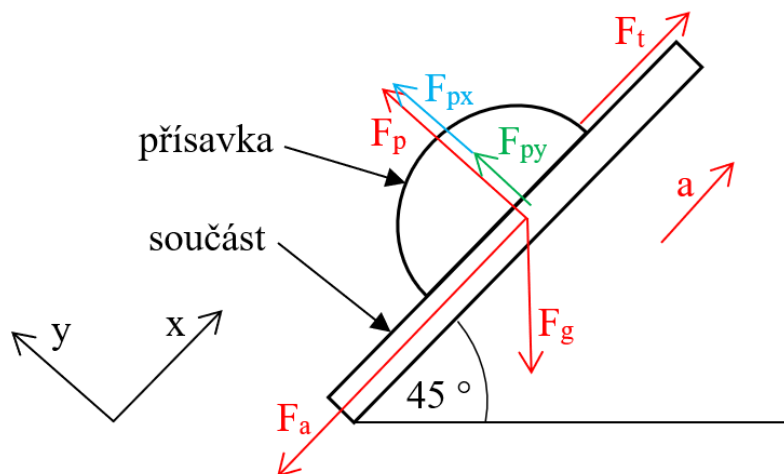
Volba efektoru v pracovišti určeném k manipulaci je klíčovou záležitostí. Dostatečná tuhost celé soustavy zaručuje přesnou paletizaci. Na druhé straně příliš hmotný efektor omezuje celkovou dynamiku robota při pohybu. Tudíž je nasnadě přistupovat k řešení kompromisně. Ze zadání (kapitola 4) vyplývá, že k manipulaci je určená součást – plech s hladkým a nečlenitým povrchem. Proto lze s výhodou pro uchopení použít pneumatické přísavky vytvářející podtlak. Jako dodavatele pneumatických komponent volím známou společnost Festo vzhledem k dobrým zkušenostem zákazníka. Pro samotnou konstrukci efektoru volím modulární díly od společnosti Aluteckk.

Hlavní konstrukci rámu efektoru tvoří profily *Modul 30* o délce 250 mm. O připevnění k robotu se stará prototypová přírubová deska. Konstrukce části k připevnění do flange robota je zkonstruována dle technické dokumentace robotů. Pro připevnění přísavek k rámu efektoru se stará součást *úchyt*, jež je přestavitelná. Tudíž lze měnit rozteč přísavek v jednom směru (Obr. 17) a docílí se tak větší variability celého efektoru. O celou tuhost systému se starají fixační komponenty od zmíněné firmy Aluteckk. Veškeré součásti (mimo pneumatických dílů) jsou vyrobeny z hliníkových polotovárů. Vzhledem ke způsobu použití efektoru nebylo nutné provedení pevnostních výpočtů.



Obr. 17 Měnitelná rozteč efektoru

Společnost Festo nabízí několik typů přísavek dle způsobu použití. Pro rovný kovový povrch je určená standardní kruhová přísavka [16]. Přísavku umístím do každého rohu součásti, tudíž jsou za potřeby celkem čtyři pro jeden efektor. K určení průměru přísavky je zapotřebí znát potřebnou přídržnou sílu „ F_p “. Největší teoretickou hodnotu vyžaduje manipulace se součástí u skluzu, jelikož se efektor natačí z horizontální polohy do úhlu 45° , viz Obrázek 18. Použitý vztahy (6.1 – 6.7) vychází z Newtonových zákonů a rovnice rovnováhy.



Obr. 18) Nejrizikovější sklon efektoru

$$\sum F_x = 0 \quad (6.1)$$

$$\vec{F}_a + \vec{F}_g + \vec{F}_t = 0 \quad (6.2)$$

$$-F_a - F_g \cos 45^\circ + F_t = 0 \quad (6.3)$$

$$-m \cdot a - m \cdot g \cdot \cos 45^\circ + F_t = 0 \quad (6.4)$$

$$F_t = m \cdot a + m \cdot g \cdot \cos 45^\circ$$

$$F_t = 2 \cdot 5 + 2 \cdot 9,81 \cdot \cos 45^\circ = 23,87 \text{ N}$$

$$\sum F_y = 0 \quad (6.5)$$

$$\overrightarrow{F_{py}} + \overrightarrow{F_g} = 0 \quad (6.6)$$

$$F_{py} - F_g \cdot \sin 45^\circ = 0 \quad (6.7)$$

$$F_{py} = m \cdot g \cdot \sin 45^\circ$$

$$F_{py} = 2 \cdot 9,81 \cdot \sin 45^\circ = 13,87 \text{ N}$$

kde F_a — reakční síla od zrychlení „a“ [N],
 a — zrychlení přísavky, resp. robota [$\text{m} \cdot \text{sec}^{-2}$],
 F_g — tíhová síla [N],
 F_t — třecí síla [N],
 m — hmotnost součásti [kg],
 g — gravitační konstanta [$\text{m} \cdot \text{sec}^{-2}$].

Přidrzná síla se skládá ze dvou dílčích sil (vztah 6.9). Velikost „ F_{py} “ značí potřebnou sílu k překonání části síly tíhové, jež je ve směru normálového vektoru součásti. Síla „ F_{px} “ zabraňuje pomocí smykového tření pohybu součásti ve směru x, viz vztah (6.8). Koeficient tření $f = 0,5$ je výrobcem doporučován v případě, že součást je vyrobena z kovového nemastného materiálu [16]. Celková přídržná síla se vypočítá dle vztahu níže uvedených.

$$F_t = F_{px} \cdot f \quad (6.8)$$

$$F_{px} = \frac{F_t}{f}$$

$$F_{px} = \frac{23,87}{0,5} = 47,75 \text{ N}$$

$$F_p = F_{px} + F_{py} \quad (6.9)$$

$$F_p = 47,75 + 13,87 = 61,62 \text{ N}$$

Nyní lze z celkové požadované přídržné síly vypočítat dle vztahu potřebnou sílu na jednu přísavku, tj. „ F_{pj} “.

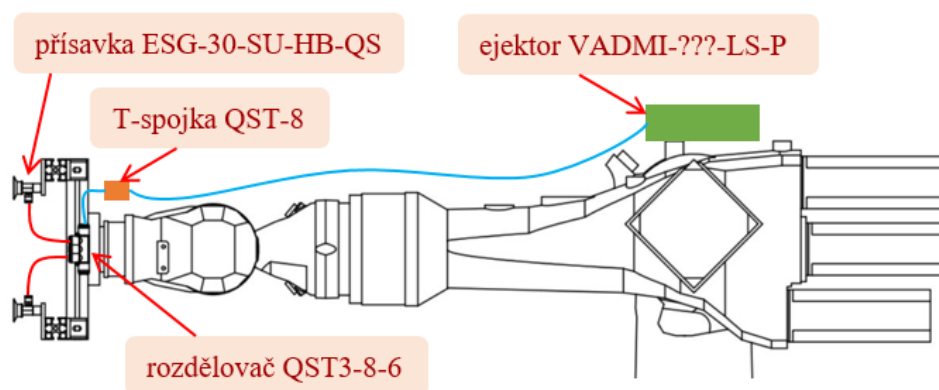
$$F_{pj} = \frac{F_p \cdot k}{n} \quad (6.10)$$

$$F_{pj} = \frac{61,62 \cdot 2}{4} = 30,81 \text{ N}$$

kde k — součinitel bezpečnosti [–],
n — počet přísavek na efektoru [–].

Z katalogu (Příloha 1) volím průměr přísavky s nejbližší vyšší hodnotou přídržné síly, tzn. $D_p = 30 \text{ mm}$. Tudíž se jeví jako nejlepší přísavka s typovým označením **ESG-30-SU-HB-QS** o objemu $V_p = 0,867 \text{ cm}^3$.

Dalším krokem je volba ejektoru, který se umístí těsně nad kloub třetí osy robota. Cílem je zvolit takový ejektor, jenž lze použít pro oba roboty i za cenu mírného předimenzování. Ve výpočtech se použije pouze robot KR 16-2 z důvodu delšího ramene oproti KR 5 ARC. Celkový pneumatický systém od přísavky po ejektor jsem navrhl dle Obrázku 19. Z výstupního místa ejektoru vede hadice PUN-8x1 do T-spojky QST-8, kde se hadice stejného typu rozdělí do dvou rozdělovačů QST3-8-6. Následně z každého rozdělovače vedou dvě hadice typu PUN-6x1 do dvou přísavek 30-SU-HB-QS.



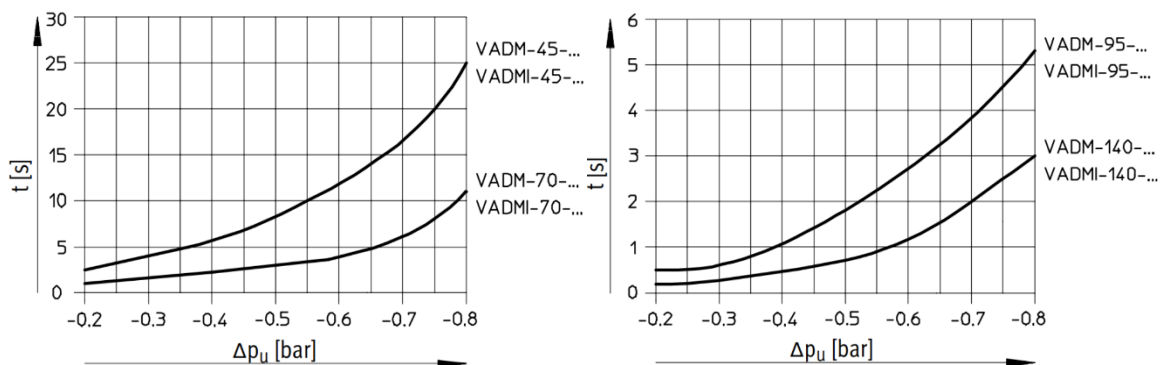
Obr. 19) Pneumatický systém manipulačního robota

Pro nutný výpočet k výběru vhodného efektoru je třeba znát celkový objem vzduchu „ V_c “, nacházející se v soustavě. Tabulka níže prezentuje jednotlivé objemy všech komponentů v pneumatické soustavě. Jednotlivé hodnoty byly včteny z katalogu a hadice dopočítány z modelu v programu Inventor Professional.

Tab 3) Objemy jednotlivých komponent [16]

komponent	typ	symbol	hodnota	jednotka
přísavka (s držákem)	30-SU-HB-QS	V1	1,513	[cm ³]
T-spojka	QST-8	V2	1,083	[cm ³]
rozdělovač	QST3-8-6	V3	1,12	[cm ³]
hadice (přísavka - rozdělovač)	PUN-6x1	V4	3,544	[cm ³]
hadice (rozdělovač - T-spojka)	PUN-8x1	V5	2,545	[cm ³]
hadice (rozdělovač - ejektor)	PUN-8x1	V6	39,881	[cm ³]

Volba vhodného efektoru se odvíjí od doby vysátí vzduchu z celé soustavy k docílení změny tlaku $\Delta p_u = -0,8 \text{ bar}$. Výrobce je doporučena doba odsátí vzduchu $t_e < 0,5 \text{ sec}$. Grafy na Obrázku 20 prezentují dobu vysátí vzduchu o objemu jednoho litru na určitou změnu tlaku. K přepočtu času odsátí vzduchu pro požadovaný objem vzduchu se provede vztahem 6.12, resp. 6.14.



Obr. 20 Doba odsátí jednoho litru vzduchu dle typu ejektoru [16]

$$V_c = 4 \cdot V_1 + V_2 + 2 \cdot V_3 + 4 \cdot V_4 + 2 \cdot V_5 + V_6 \quad (6.11)$$

$$V_c = 4 \cdot 1,513 + 1,083 + 2 \cdot 1,12 + 4 \cdot 3,544 + 2 \cdot 2,545 + 39,881$$

$$V_c = 68,522 \text{ cm}^3$$

$$t_{e70} = V_c \cdot \frac{t_{70}}{1000} \quad (6.12)$$

$$t_{e70} = 68,522 \cdot \frac{11}{1000} = 0,754 \text{ sec}$$

$$t_{e70} > 0,5 \quad (6.13)$$

$$t_{e95} = V_c \cdot \frac{t_{95}}{1000} \quad (6.14)$$

$$t_{e95} = 68,522 \cdot \frac{5,2}{1000} = 0,356 \text{ sec}$$

$$t_{e95} < 0,5 \quad (6.15)$$

$$t_e = t_{e95} = 0,356 \text{ sec} \quad (6.16)$$

kde t_{70} — doba odsátí jednoho litru vzduchu k docílení změny tlaku $\Delta p_u = -0,8$ bar pomocí ejektoru VADMI-70 [sec],

t_{e70} — doba odsátí vzduchu o objemu „ V_c “ k docílení změny tlaku $\Delta p_u = -0,8$ bar pomocí ejektoru VADMI-70 [sec].

t_{95} — doba odsátí jednoho litru vzduchu k docílení změny tlaku $\Delta p_u = -0,8$ bar pomocí ejektoru VADMI-95 [sec].

t_{e95} — doba odsátí vzduchu o objemu „ V_c “ k docílení změny tlaku $\Delta p_u = -0,8$ bar pomocí ejektoru VADMI-95 [sec].

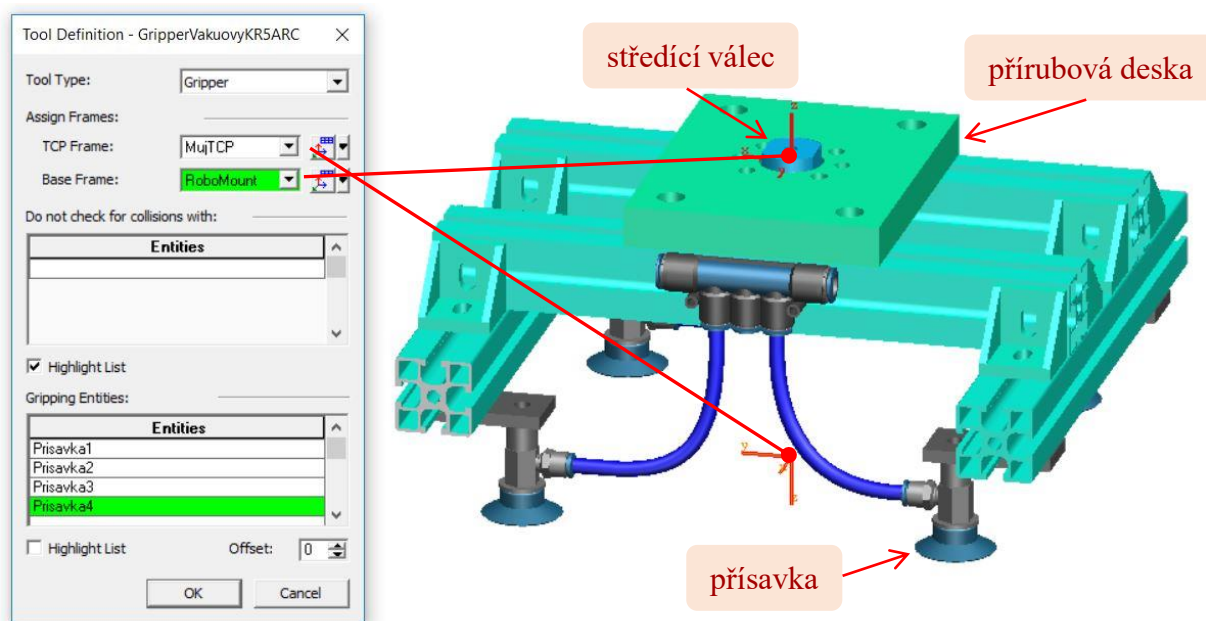
Z výpočtu plyne, že jako nejvhodnější efektor se jeví VADMI-95, resp. **VADMI-95-LS-P**.

6.3.2 Implementace do SW PS

Pokud se v projektu nachází prototypový efektor, je potřeba řádně definovat jeho základní části. K tomu je určena funkce *Modeling* → *Kinematic Device* → *Tool Definition*, jež umožňuje definovat objekt jako efektor. Následný efektor může být posléze připojen k robotu, aby mohl provádět jednotlivé úkony. Možnost *Tool Type* se navolí dle typu efektoru:

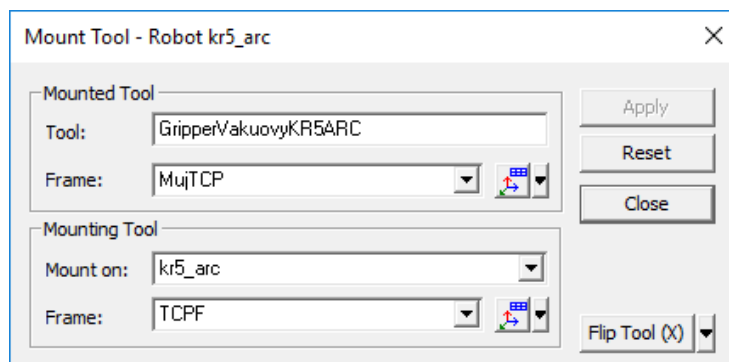
- ✓ *Servo Gun* — svařovací kleště se servo pohonem,
- ✓ *Pneumatic Servo Gun* — pneumatické svařovací kleště,
- ✓ *Gun* — obecné svařovací kleště nespádající do kategorií výše uvedených,
- ✓ *Paint Gun* — malířské pistole,
- ✓ *Gripper* — veškeré manipulační efekty [10].

Pro následující položky v kategorii *Assign Frames* je vhodné si předem připravit *frame* pevně svázané s modelem, viz *MujTCP* a *RoboMount* na Obrázku 21. *TCP Frame* označuje středový bod úchopu efektoru v rovině s koncem přísavek. Tento bod se bude při tvorbě robotických operací ztotožňovat s bodem na povrchu právě uchopované součásti. Pokud není uvedeno jinak, souřadnice lokací robota se vždy vztahují k *TCP*. *Base Frame* slouží k ukotvení efektoru do flange robota. Nachází se na povrchu přírubové desky, jež obsahuje otvory pro šrouby a středícího kolíku k upevnění efektoru. Středící válec se do flange robota zapouští, tudíž jeho horní povrch nelze považovat za spojovací rovinu. Jednotlivé přísavky jsou v kontaktu s uchopovanou součástí. Proto se klasifikují jako *Gripping Entities*.



Obr. 21) Definování nástroje v PS

Připravený efektor se připojí k robotu pomocí *Robot* → *Tool and Device* → *Mount Tool*, viz Obrázek 22. Kategorie *Mounted Tool* vyžaduje informace o nástroji. Položka *Tool* vyžaduje *resource* klasifikovaný jako efektor. *Frame* žádá místo pro ukotvení na flange robota, tj. *MujTCP*. *Mount on* žádá *resource* reprezentujícího robota. Následný *Frame* je bod ve středu konce flange, tj. *TCPPF*. Po stiknutí *Apply* se efektor připojí na robota. Pokud ukotvení efektoru není správné, může za to špatná orientace některého z *frame*. Funkce *Flip Tool* napraví tuto skutečnost. Nicméně je vhodnější samotný *frame* znovu přeorientovat v *Modeling Scope*.



Obr. 22) Dialogové okno pro připevnění efektoru k robotu

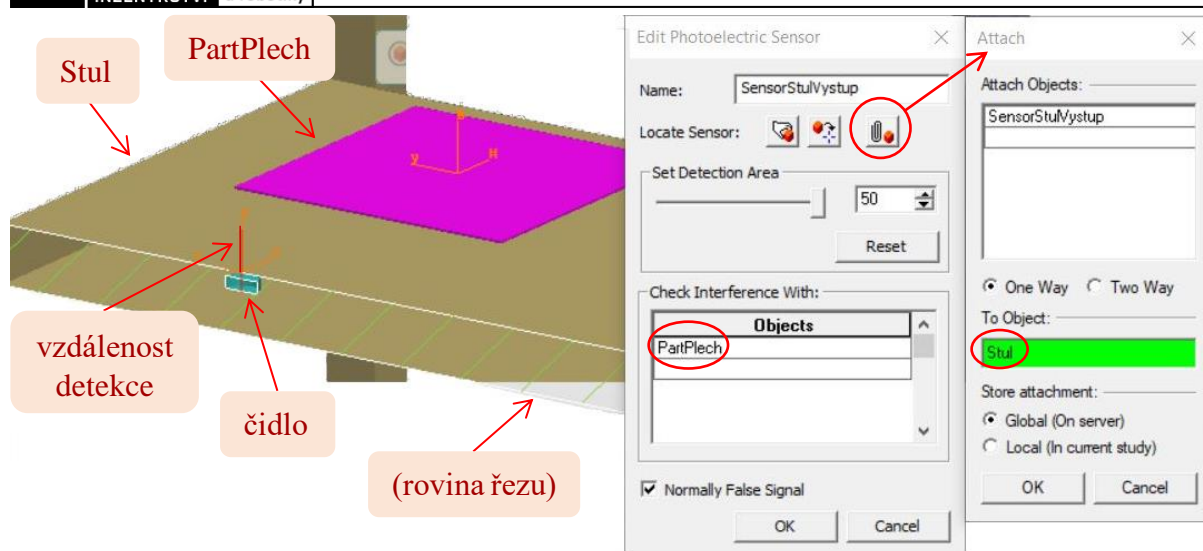
6.4 Senzory

V programu PS lze nalézt několik typů senzorů. V simulaci je fyzicky zastupuje volitelný 3D objekt a výstupem je vždy jeden signál informující o stavu konkrétního senzoru. Celkem pět typů senzorů má následující funkce:

- *Joint Distance Sensor* — snímač pro kloubové spojení. Výstupem je číselná proměnná udávající vzdálenost nebo úhel aktuální polohy kloubu od polohy výchozí.
- *Joint Value Sensor* — snímač pro kloubové spojení. Změna stavu proběhne, jakmile se kloub dostane do předem stanovené pozice, popř. do pozice v rámci stanovené tolerance. Výstupem může být pouhý puls, obdélníkový signál, náběžná hrana nebo sestupná hrana signálu.
- *Proximity Sensor* — snímač detekující přítomnost stanovených objektů v určeném rozsahu. Detekční rozsah se rovná poloměru koule, jež značí detekční prostor senzoru.
- *Photoelectric Sensor* — snímač detekující přítomnost stanovených objektů, jež se dostanou do kolize s paprskem, resp. vektorem. Délka paprsku je volitelná.
- *Property Sensor* — snímač detekující vlastnost součásti nacházející se ve stanovené trojrozměrné detekční zóně [10].

Senzory umístěné v periferiích zadaného projektu jsou určeny pouze k detekci přítomnosti součásti, tudíž výstupem ze senzoru je booleovská proměnná. Reálně se jedná o indukční nebo kapacitní senzory. V programu PS se k tomuto typu použije *Control* → *Sensor* → *Sensors* → *Create Photoelectric Sensor* i za předpokladu, že se nejedná o fotoelektrické snímače. Jsou vhodné i z hlediska náročnosti na výpočet během simulace. Modelová interpretace senzoru pro účely simulace není důležitá. Je pouze nutno dodržet umístění snímacího paprsku dle následného umístění senzoru v reálném pracovišti.

Modelové ztvárnění čidla v simulaci je válec o průměru 30 mm a výšce 10 mm (v tomto případě není důležité), viz Obrázek 23. Čidlo je vnořeno do modelu stolu a permanentně s ním spjato pomocí funkce *Attach*. Detekční vzdálenost je ponechána na výchozí hodnotě 50 mm a jako jediný objekt, jenž může být senzorem detekován, je zvolená manipulovaná součást *PartPlech*. V opačném případě by senzor stále detekoval přítomnost stolu, jelikož je čidlo v něm zapuštěno, což je nežádoucí.



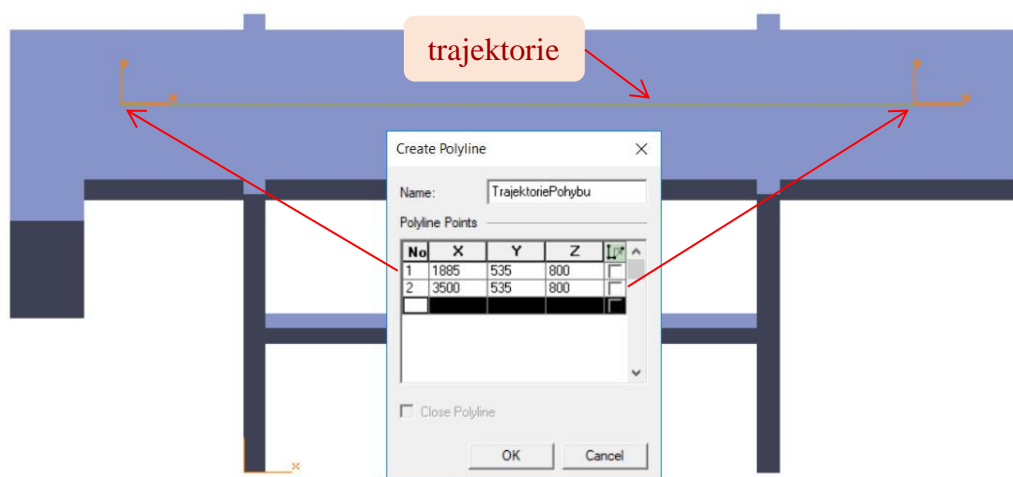
Obr. 23) Definování senzoru v PS

6.5 Dopravník

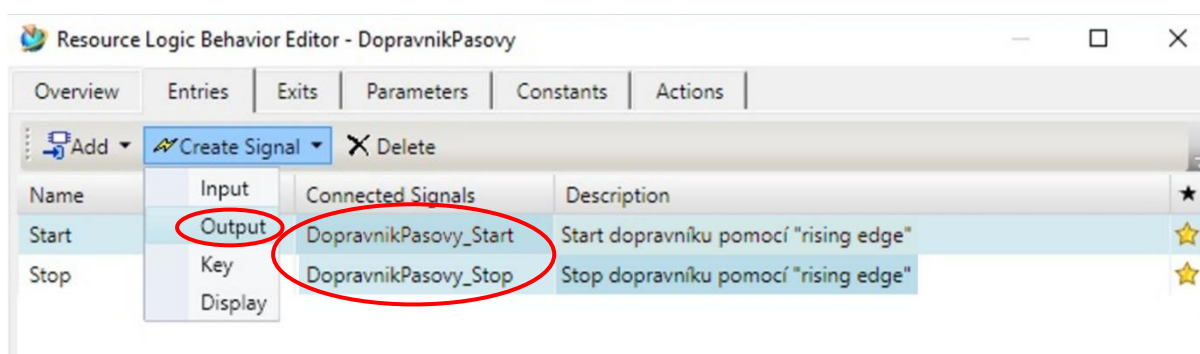
K definování pásového dopravníku je potřeba předem nachystaná trajektorie pohybu, tj. úsečka od vstupního bodu k výstupnímu. Pomocí *Modeling* → *Geometry* → *Curves* → *Create Polyline* se úsečka vytvoří. Po otevření dialogového okna *Create Polyline* postačí kliknout na vstupní a výstupní předem vytvořené *frame*, viz Obrázek 24. V následně otevřeném dialogovém okně *Control* → *Conveyor* → *Edit Conceptual Conveyor* se tato úsečka vloží do položky *Curve*. Zbývající položky se ponechají ve výchozích hodnotách vyjma *Max Speed*, kde se nastaví rychlost dopravníku. Zvolil jsem rychlost $494 \text{ mm} \cdot \text{sec}^{-1}$, jež jsem vypočítal z jmenovitých otáček pohonu v kapitole 4.2.

K ovládání vytvořeného pásového dopravníku jsou potřeba signály. Program PS má předem nachystané řízení pro dopravníky, a proto stačí zakliknout *Start* a *Stop* v dialogovém okně *Control* → *Conveyor* → *Edit Conveyor LB*. V zadaném projektu se neuvažuje o změně směru pásového dopravníku, proto se vstup k tomuto účelu nevytváří. Ke vstupům se vytvoří příslušné signály pro ovládání pomocí externí automatiky. Po označení konkrétního řádku se vytvoří signál podle Obrázku 25. I když se jedná o vstupy do dopravníku, jsou to výstupní signály (bráno z pohledu PLC). Sloupec *Description* je volitelný a není nutné zde nic psát. Vytvořené signály se automaticky pojmenují, viz Obrázek 25. Pokud je nutné dodržet určitě směrnice v názvosloví, lze jim změnit název v *View* → *Screen Layout* → *Viewers* → *Signal Viewer* kliknutím na požadovaný řádek a stisknutím klávesy F2 pro aktivaci přejmenování.

V posledním kroku je třeba určit součást, jež se bude po dopravníku pohybovat. Kliknutím na příslušný objekt v grafickém okně nebo v *Object Tree* se aktivuje funkce *Control* → *Conveyor* → *Define Conveyable Part*. Položka *Conveying Frame* žádá souřadnice bodu, jež se bude po výše zmíněné trajektorii dopravníku pohybovat. K tomu poslouží například předem vytvořený *frame* na rubu součásti.



Obr. 24) Trajektorie dopravníku



Obr. 25) Vytvoření signálů pro ovládání dopravníku

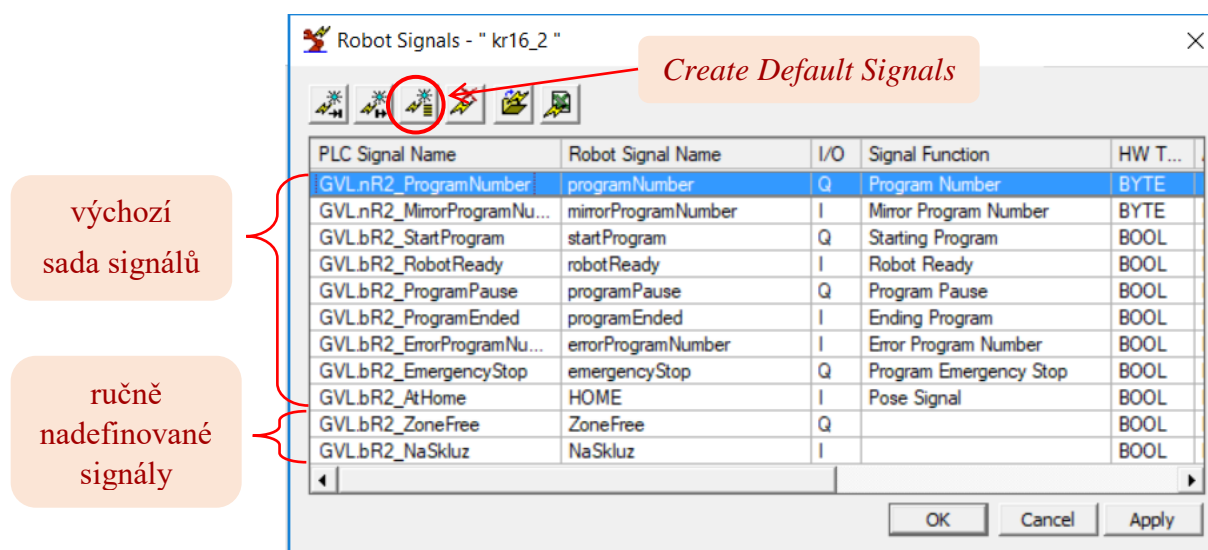
6.6 Skluz

Vytvoření gravitačního dopravníku, resp. skluzu je mírně komplikované, jelikož není na něj program PS zcela připraven. První, jednodušší způsob je nahrazení „sklouznutí“ součásti pomocí *Object Flow Operation*. Nastaví se výchozí a cílová pozice, doba trvání přesunu a „sklouznutí“ součásti bude bráno jako jedna z operací. Nicméně tento způsob může způsobit v budoucnu problémy. Zejména ve virtuálním zprovoznění, kde jsou veškeré operace řízeny logickými událostmi.

Druhý způsob staví na principu definování klasického dopravníku. Opět je potřeba vytvořit trajektorii ze vstupního místa k výchozímu pomocí *Modeling* → *Geometry* → *Curves* → *Create Polyline*. Následně je potřeba otevřít dialogové okno *Control* → *Conveyor* → *Define Conveyor* a vytvořenou úsečku vložit do položky *Curve*. V možnosti *Max Speed* → *Linear* se $300 \text{ mm} \cdot \text{sec}^{-1}$ jeví jako optimální rychlost pro gravitační spád pod úhlem 45° . Důležitým krokem je vytvoření v *Control* → *Conveyor* → *Edit Conveyor LB* pouze vstup *Start* bez *Stop*. Díky tomu bude brán skluz jako neustále pohybující se klasický dopravník. Jelikož není potřeba řídit skluz pomocí PLC, nevytváří se signál pro vstup *Start*.

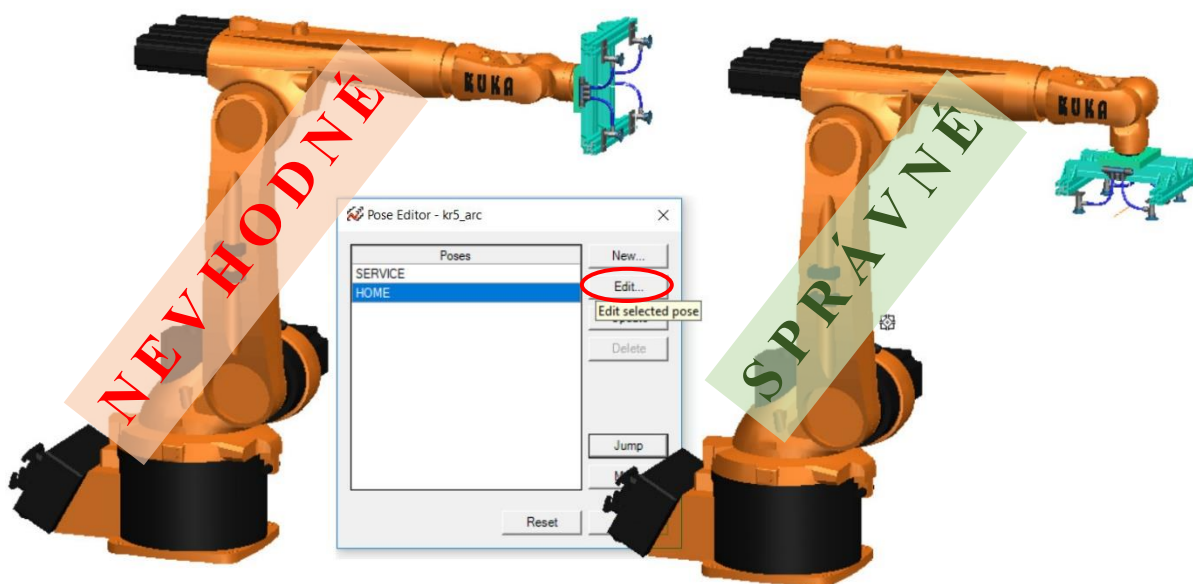
6.7 Příprava robotů

Program PS obsahuje výchozí sadu signálů pro řízení robota externí automatikou. Tato sada signálů je pro zadaný projekt dobrým základem. Vygenerování signálu se provede v dialogovém okně v nabídce robota (pravým kliknutím na robota v grafickém okně) → *Robot signals and Modules* → *Robot Signals*. Po stisknutí tlačítka *Create Default Signals* se vygeneruje sada signálů, viz Obrázek 26. Jelikož v zadaném projektu může dojít v určitém místě ke kolizi obou robotů, je nutné vytvořit navíc jeden vstupní a výstupní signál pro oba roboty. Díky těmto signálům lze v externí automatice sestavit algoritmus hlídající vzájemný střet. Více informací lze naléznout v kapitole 7. Soubor signálů lze přejmenovat dle zvolené směrnice pro názvosloví v *View* → *Viewers* → *Signal Viewer*.



Obr. 26) Řídící signály robota

Dalším krokem je vytvoření základních pozic obou robotů. V zadaném projektu postačí pouze domovská poloha. Předem definovaná *Home* pozice robota není ideální. Díky nataženému ramenu u efektoru hrozí nebezpečí singulární polohy. Změna lze provést označením robota v grafickém okně nebo v *Object Tree* a otevřít dialogové okno *Modeling* → *Kinematic Device* → *Pose Editor*. Následným stisknutím tlačítka *Edit* (Obr. 27) se v dialogovém okně *Edit Pose* navolí požadované natočení jednotlivých kloubů. V mém případě postačí změnit hodnotu natočení kloubu páté osy z 0 ° na 90 °. Takto lze nastavit další pozice jako například poloha pro servisní operace. Pokud je tlačítko *Edit* neaktivní (zašedlé), musí se robot odemknout pro modelování (*Modeling* → *Scope* → *Set Modeling Scope*).



Obr. 27) Změna *Home* pozice robota

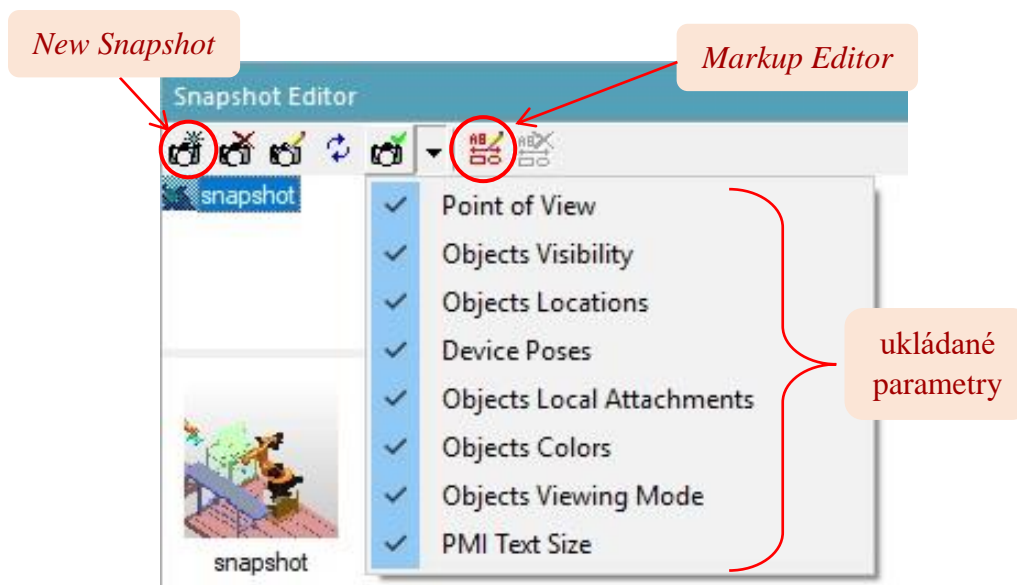
6.8 Tvorba robotických operací

Před započítím tohoto úkolu je vhodné si nejprve uložit výchozí polohu robotů a pozici manipulované součásti pomocí screenshot funkce programu PS. Další *Snapshots* mohou sloužit pro uložení pozice manipulované součásti k tvorbě dalších robotických operací, jež navazují na operace předchozí. Jiný *snapshot* může sloužit k uložení viditelnosti (zobrazeno nebo nezobrazeno) jednotlivých entit.

Uložení se provede v panelu *View* → *Screen Layout* → *Viewers* → *Snapshot Editor* pomocí tlačítka *New Snapshot*, viz Obrázek 28. Může se zdát zavádějící, že vyfocení grafického okna uloží pozice periférií, ale funkce *Snapshot* v PS funguje komplexněji. Vytvořený snímek ukládá:

- ✓ aktuální pohled,
- ✓ nastavenou viditelnost entit v grafickém okně,
- ✓ polohu objektů v layoutu,
- ✓ pozici objektů (natočení ramena robota, otevřený chapač apod.),
- ✓ barvu objektů
- ✓ a další.

Další užitečnou funkcí v dialogovém okně *Snapshot Editor* je *Markup Editor*, v němž lze uložený pohled editovat ve smyslu přidání popisků, zvýraznění lokace pomocí geometrických tvarů apod. Používá se běžně pro tvorbu reportů nebo prezentací k projektu. Načtení *snapshotu* a vrácení všech změn provedených po jeho uložení, se provede dvojklikem na vybraný *snapshot*.

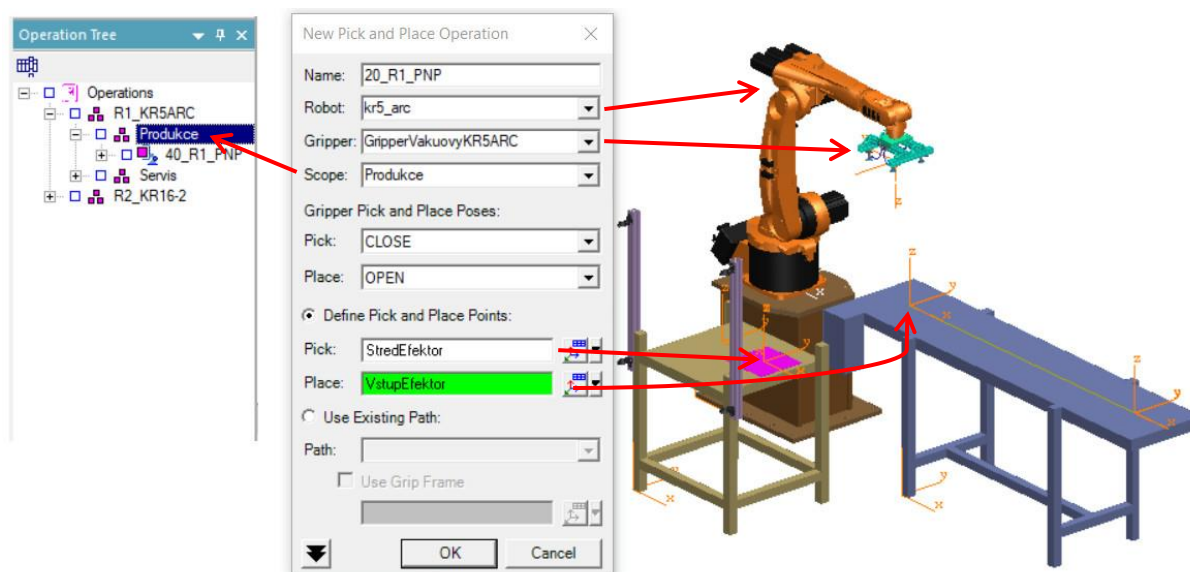


Obr. 28) Dialogové okno *Snapshot Editor*

K tvorbě programu pro roboty v zadaném projektu postačí druh operace *Pick and Place Operation* a *Generic Robotic Operation*, jelikož roboty provádí pouze manipulační operace a případně se provede najetí do definované polohy pro servisní činnost.

Nejprve je vhodné si vytvořit skupiny operací, jež pomůže logicky rozřadit jednotlivé operace. Vytvoření probíhá v *Operation* → *Create Operation* → *New Operation* → *New Compound Operation*. *Name* značí název skupiny a *Scope* určuje, do jaké úrovně stromové struktury (tvořené ze skupin operací v *Operation Tree*) se nová operace zařadí. Výchozí možnost *Operation Root* je nejvyšší úroveň.

Přesun součásti ze stolu na dopravník pomocí robota se provede následovně. V dialogovém okně (Obr. 29) *Operation* → *Create Operation* → *New Pick and Place Operation* se zvolí požadované objekty z *Object Tree* (lze i kliknutím v grafickém okně) pro položky *Robot* a *Gripper*. *Scope* určuje předem vytvořenou skupinu, do níž bude tato operace patřit. Kategorie *Gripper Pick and Place Poses* není důležitá, jelikož se jedná o pneumatický efektor a během úchopu součásti se neprovádí žádný mechanický pohyb (např. sevření chapače). *Pick* žádá výběr *frame*, jenž se při uchopení součásti pohybem robota ztotožní s *frame* definující TCP. Jedná se například o *frame* zastupující střed horní strany manipulované součásti. Obdobně se řeší *Place*. Je třeba mít na paměti, že *frame* pro *Place* musí být v určité výšce (například tloušťku manipulovaného plechu) nad odkladovou plochou.

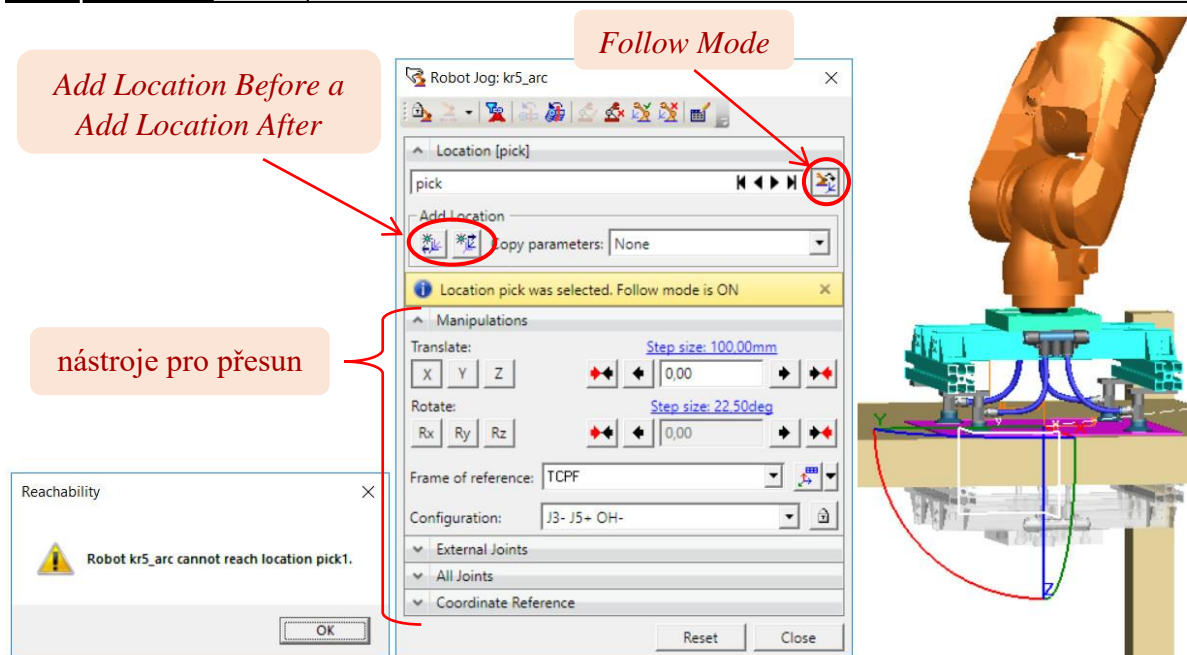


Obr. 29) Pick and Place operace

Po potvrzení tlačítkem *OK* se operace *Pick and Place* vytvoří s nejkratší možnou cestou (úsečka mezi dvěma lokacemi). Ve většině případů takto jednoduché řešení není vhodné a je nutné vytvořit průchodové lokace, jelikož může vzniknout během cesty kolize. Označením lokace *pick* v *Operation Tree* a následným kliknutím na příkaz *Operation* → *Edit Path* → *Manipulate Location* se otevře dialogové okno *Robot Jog* (Obr. 30) a robot se přemístí do označené lokace.

Může nastat chyba (Obrázek 30), jež je zapříčiněna z důvodu: cílové lokace mimo pracovní obálku robota, robot se snaží uchopit součást ze spodní strany (Obr. 30) nebo je nevhodně přetočený efektor kolem osy *z* (osa kolmá na uchopovanou součást). Ve všech případech robot zůstane v původní pozici a v cílové lokaci (např. *pick*) se vytvoří fiktivní duplikát efektoru šedé barvy, jenž demonstruje aktuálně nastavenou (nedosažitelnou) uchopovací pozici. Chybu nelze opravit ve výše zmíněném *Manipulate Location*, ale pomocí *Operation* → *Edit Path* → *Flip Locations*. Při tomto úkonu je důležité mít označenou chybnou lokaci v *Operation Tree*. Druhou možností je odstranění celé operace a vhodné pootočení *TCP frame* nebo *frame* značící místo pro uchopení. Pokud se ale lokace nachází mimo pracovní obálku robota, je zapotřebí lokaci přesunout do dosažitelné vzdálenosti.

Po úspěšném otevření *Manipulate Location* bez vykazované chyby se robot přemístí do označené lokace. Pro přemístění aktuální lokace se ponechá *Follow Mode* zapnutý. Pro vytvoření nové je nutné tento mód vypnout. Během vypnutého *Follow Mode* se pomocí translace v jednotlivých osách rameno robota přesune na zvolené místo v prostoru. Funkcemi *Add Location Before* nebo *Add Location After*, jež se nachází přímo v otevřeném okně *Robot Jog*, se ze zmíněného místa vytvoří průchodová lokace. Pro změnu chronologického pořadí lokací v právě editované operaci se používá *Operation* → *Edit Path* → *Shift Location Back* nebo *Shift Location Forward*. V prvotních návrzích je příhodné nastavit (pro každou operaci) první a poslední lokaci domovskou pozici robota. Označením robota v grafickém okně a následným stisknutím klávesy *Home* se robot přesune právě do této pozice. Z aktuální pozice robota se vytvoří lokace do označené operace pomocí *Operation* → *Add Location* → *Add Current Location*.



Obr. 30) Úprava lokací

vlevo — chyba nevhodné lokace; střed — Robot Jog; vpravo — následek chyby

V neposlední řadě je vhodné si lokace řádně zorientovat (často rotace kolem osy Z). Bez řádné orientace se bude efektor během přesouvání neúčelně přetáčet, což je nežádoucí. Nejdříve se zvolí jedna lokace jako výchozí. Následně se využije funkce *Operation* → *Edit Path* → *Align Locations*. V dialogovém okně se do kolonky *Align selected locations to* zvolí výše zmíněná výchozí lokace a do kolonky *Selected locations* se z *Operation Tree* navolí veškeré lokace určené k řádnému zorientování.

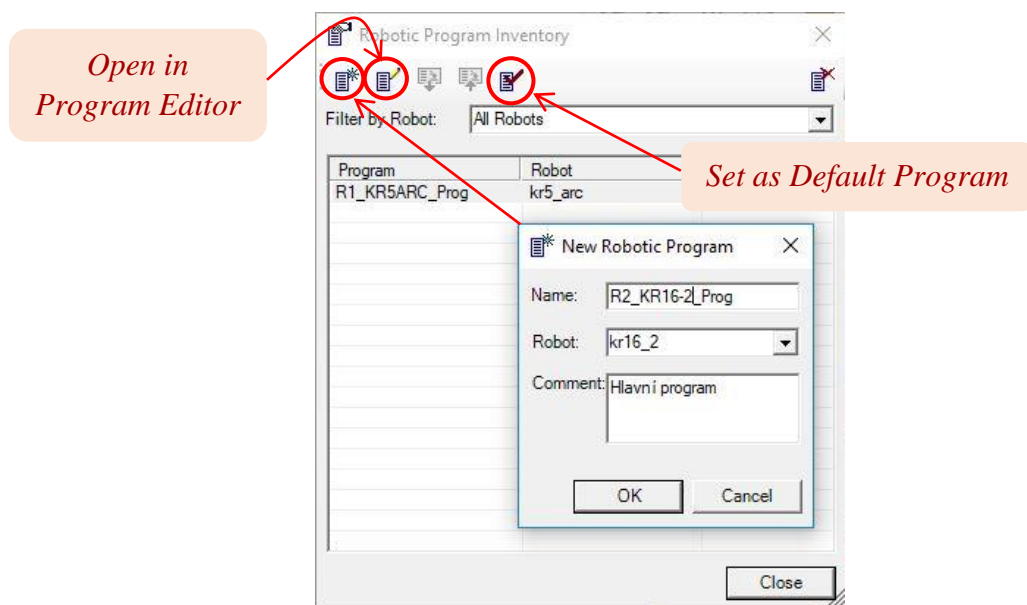
Pro servisní operace je vhodné použít typ *Operation* → *Create Operation* → *New Operation* → *New Generic Robotic Operation*. Daný typ operace umožňuje robotu obecný pohyb z místa na místo. V zadaném projektu je třeba definovat operaci k natočení ramene tak, aby mohla údržba snadno očistit přísavky. Funkce *New Generic Robotic Operation* vytvoří prázdnou operaci. Přidání lokace lze například pomocí dialogového okna *Robot* → *Reach* → *Robot Jog*, kde se nastaví požadovaná poloha. Následně se označí zmiňovaná operace v *Operation Tree* a pomocí *Operation* → *Add Location* → *Add Current Location* se přidá lokace podle aktuálního umístění robota.

Je vhodné si u všech operací překontrolovat, zdali obsahují informaci, s jakou součástí se bude kolaborovat. Pravým kliknutím na příslušnou operaci v *Operation Tree* se zobrazí nabídka možností a zvolením *Operation Properties* se otevře dialogové okno. V kartě *Products* je nutné navolit manipulovanou součást jako *Product Instances*. Přidá se po kliknutí na součást v *Object Tree* nebo v grafickém okně, viz Obrázek 33.

Pro další práci s vytvořenými operacemi je nutné vytvořit programy pro roboty. Jedná se o hlavní výstup z programu PS, kde se nadefinované pohyby v operacích pomocí postprocesoru překonvertují do příslušného jazyka robotu (např. jazyk KRL pro robot KUKA). Exportovaný soubor lze následně nahrát přímo do reálného robota, čímž je možné dosáhnout rychlejšího uvedení do provozu celého pracoviště. Pomocí *Robot* → *Program* → *Robotic Program Inventory* se otevře dialogové okno, kde stisknutím tlačítka *Create New Program* se nadefinuje nový program pro každého robota zvlášť, viz Obrázek 31. Následně se nastaví programy jako

výchozí příkazem *Set as Default Program*, čímž se font popisující název programů ztuční. Příkazem *Open in Program Editor* se právě označený program přenese do *View -> Viewers -> Path Editor*. Zde se provádí vložení jednotlivých operací do programu příslušného robota. Označením operace (nikoliv *Compound Operation*) a stisknutím tlačítka *Add Operation to Program* v *Path Editor* se provede vložení. Pokud je tlačítko zašedlé, pravděpodobně je operace určena pro jiného robota. Pro virtuální zprovoznění je důležité přiřadit číslo pro každou operaci ve sloupci *Path #*, kterým bude operace vyvolána pro vykonání. Pokud atribut *Path #* nelze v *Path Editor* nalézt, lze jej přidat pomocí tlačítka *Customize Columns*, viz Obrázek 32.

Takto přiřazené operace ke konkrétnímu robotu je vhodné odsimulovat a překontrolovat dráhu, časy apod. K tomuto účelu slouží funkce *Auto Teach*. Po její aktivaci a stisknutí tlačítka *Play Simulation Forward* se začnou simulovat veškeré operace přidané do programu robota. Průběh simulace vytváří kontrolér, jenž lze nastavit v *Robot -> Setup -> Controller Settings*.



Obr. 31) Dialogové okno *Robotic Program Inventory*

Po vytvoření operace, se všem lokacím nastaví výchozí atributy. Jedná se o *Point To Point* pohyb, 100 % rychlost, přesné najetí robotem do určené lokace a další. Změny se provádí v jednotlivých sloupcích u každé lokace zvlášť v *Path Editor*. Chybějící atributy lze přidat pomocí funkce *Customize Columns*. Mezi nejdůležitější atributy (Obrázek 32) lze zařadit ze skupiny *Kuka-Krc: Zone, Speed* a *Motion*:

❖ Atribut *Motion*

Pro pohyb ve volném prostoru směrem k lokaci *Pick* nebo *Place* je vhodné použít typ pohybu *PTP*, což umožní přesun robota po dráze s nejkratším možným časem. Nicméně pro přejezd přímo do lokace *Pick* nebo *Place* se používá pohyb *LIN*, jenž zapříčiní lineární, kolmý nájezd k uchopované součásti.

❖ Atribut Speed

Rychlost se udává v procentech pro *PTP* pohyby (běžně se ponechává 100 %) a v metrech za sekundu pro *LIN* pohyby. Jelikož na konci lokace *Pick* nebo *Place* dojde ke kontaktu součásti nebo odkladové plochy s efektem, doporučuji nastavit hodnotu maximálně do $0,2 \text{ m} \cdot \text{s}^{-1}$.

❖ Atribut Zone

Atribut *Zone* určuje, jak přesně má robot danou lokaci projet. Hodnota *fine* značí, že robot najede do lokace přesně, což značně zpomalí jeho pohyb. Hodnota *fine* se používá pro klíčové lokace *Pick* nebo *Place*. K ostatním lokacím určeným pouze pro specifický průjezd prostorem se může přiřadit funkce *C_DIS* (aproximační parametr), což sníží dobu trvání robotické operace. V závislosti na položení konkrétní lokace doporučuji používat hodnoty *C_DIS* 20 mm až *C_DIS* 200 mm.

❖ Atribut OLP Commands

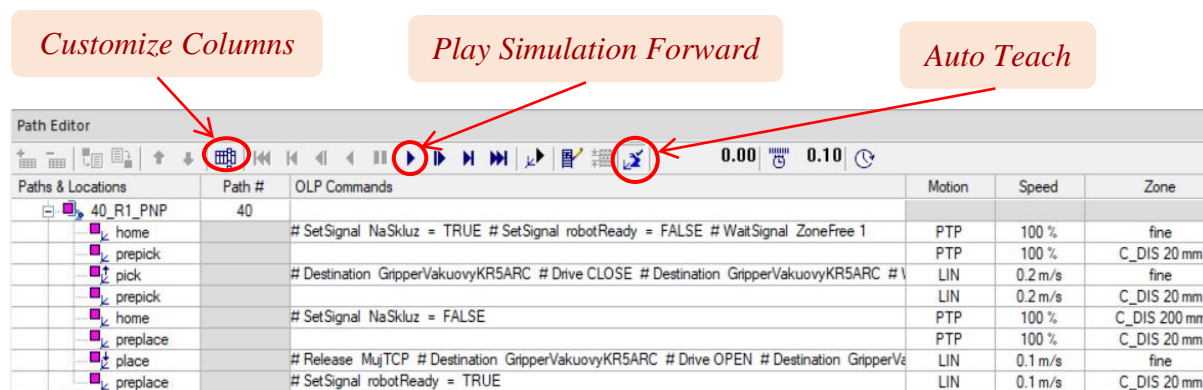
V posledním kroku se přiřadí signály konkrétního robota k jednotlivým lokacím v operaci pomocí *OLP Commands*. Výchozí signály robota, jež se dříve vygenerovali tlačítkem *Create Default Signals* v dialogovém okně *Robot Signals* (kapitola 6.7), mají svou funkci již přiřazenou s výjimkou signálu *robotReady*. Při testování se ukázalo, že pokud se robot nachází v nečinném stavu a je připraven, signál se nepřepne na hodnotu *True*. Dvojklikem na kolonku *OLP Commands* pro příslušnou lokaci se otevře dialogové okno. Tlačítkem *Add* se přidá nový příkaz. Pro zadaný projekt jsou nejdůležitější příkazy:

➤ Add → Standard Commands → Synchronization → SetSignal

Nastavení pouze vstupních signálů (z pohledu PLC) na konkrétní hodnotu (*True* nebo *False*). Použito na první a poslední lokaci v každé operaci, kde příkaz přepíná signál *robotReady* na *False* (robot začal vykonávat operaci) a *True* (robot dojel do poslední lokace a je nečinný). Obdobně se nastaví i signál *NaSkluz*. Pokud projede robot lokací, jež ohraničuje kolizní zónu s druhým robotem, nastaví se signál na hodnotu *True* a naopak.

➤ Add → Standard Commands → Synchronization → WaitSignal

Robot vyčká na konkrétní lokaci, dokud se signál nepřepne na předem nastavenou hodnotu v *OLP* příkazu. Požadovaná hodnota signálu lze nastavit pouze pomocí přirozeného čísla, tj. nula pro *False* a jedna pro *True* v případě booleovské proměnné. Mohou se použít pouze výstupní signály (z pohledu PLC). Použito na signál *ZoneFree* pro lokaci, jež ohraničuje kolizní zónu s druhým robotem.



Obr. 32) Ukázka nastavené operace 40 v *Path Editor*

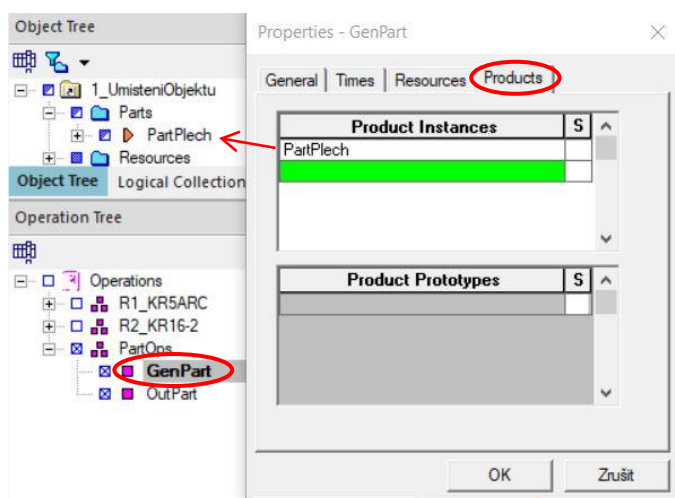
6.9 Generování součástí

Zadaný projekt simuluje pohyb manipulované součásti v pracovišti. Tudiž je potřeba vytvořit způsob (z pohledu simulace), jakým bude součást vstupovat a vystupovat z robotické buňky. K tomu existují dva způsoby:

- 1) generování a odvod součásti pomocí *Object Flow Operation*. Používá se pro vstupní nebo výstupní dopravník. Operace simuluje pohyb součásti z vnějšího prostředí do prostoru pracoviště nebo naopak.
- 2) generování a odvod součásti pomocí *Non-Sim Operation*. Používá se pro paletizace a stohy. Součást se zjeví na předem určeném místě bez simulace pohybu. Jedná se o případy, kdy jsou součásti přiváděné do pracoviště na určené místo v dávkách pomocí například operátora. Manipulované součásti vytvoří stoh na vstupním místě pracoviště. Robot odebere první součást a součást druhá se pomocí *Non-Sim Operation* zjeví na stejném místě. Tvzení platí obdobně pro odvod součásti z výstupního místa pracoviště.

Poznámka: teoreticky lze použít *Object Flow Operation* i pro druhý způsob, pokud se nastaví startovací a cílová pozice na stejný *frame*. To zapříčiní, že součást se bude pohybovat po nulové dráze, tj. pohyb začne a skončí na stejné pozici.

V zadaném projektu se manipulované součásti nacházejí ve stohu na vstupním místě a robot součást paletizuje opět do stohu na výstupní místo. Proto je vhodné použít druhý způsob. Před započítím doporučuji vytvořit novou *Compound Operation* v *Operation Tree*, jež bude obsahovat operace na generování a odvod součásti. Nejdříve se manipulovaná součást umístí na požadované místo pomocí *Relocate* nebo *Placement manipulator*. Následně se vytvoří nová operace typu *Non-Sim Operation* s volitelným názvem *GenPart* a položkou *Duration* nastavenou na 0 sekund. Pravým kliknutím na nově vytvořenou operaci v *Operation Tree* se zobrazí nabídka možností a zvolením *Operation Properties* se otevře dialogové okno. V kartě *Products* je nutné navolit manipulovanou součást jako *Product Instances*. Přidá se po kliknutí na součást v *Object Tree* nebo v grafickém okně, viz Obrázek 33. Obdobně se vytvoří i operace *OutPart* pro odvod manipulované součásti, přičemž je nutné nejdříve součást přemístit na výstupní lokaci.

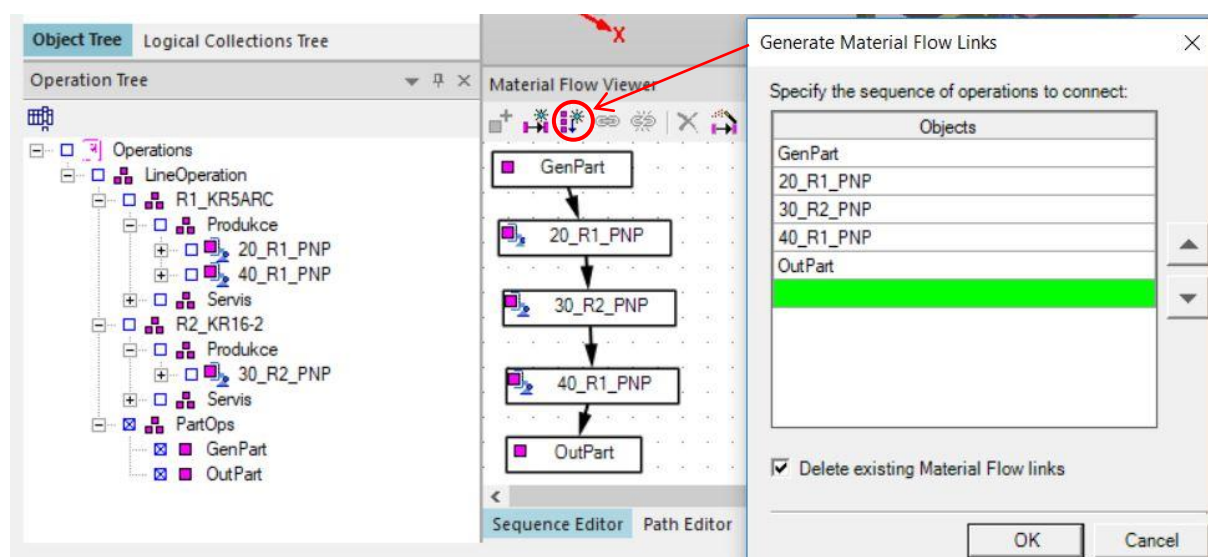


Obr. 33) Přiřazení součásti k operaci

K založeným operacím je nutné vytvořit spouštěcí signály pro jejich ovládání externí automatikou. Kliknutím na příslušnou operaci v *Operation Tree* se aktivuje funkce *Control* → *Operation Signals* → *Create Non-Sim Start Signal*, jež signál automaticky vytvoří a nadefinuje. Zobrazí se v *Signal Viewer*.

6.10 Materiálový tok

Program PS vyžaduje zadání materiálového toku. Jedná se o posloupnost operací v simulaci. Po přepnutí do logicky řízené simulace v *Home* → *Study* → *Line Simulation Mode* je možné tok materiálu nastavit ve *View* → *Screen Layout* → *Viewers* → *Material Flow Viewer*. Pro jednoduché řetězení operací doporučuji použít funkci *Generate Material Flow Links*. Po aktivaci se zobrazí dialogové okno, kam se v daném pořadí přidávají dříve vytvořené operace pomocí kliknutí na jejich název v *Operation Tree*. Materiálový tok se definuje od vstupního místa k výstupnímu, viz Obrázek 34.



Obr. 34) Definování materiálového toku

7 VIRTUÁLNÍ ZPROVOZNĚNÍ ROBOTICKÉ BUŇKY

V zadaném projektu je potřeba, aby simulaci v programu PS řídila externí automatika, respektive nadřazené PLC. PS umožňuje virtuální zprovoznění pouze v *event-based* simulaci nikoliv ve standardní *time-based* simulaci, jelikož je nutné povolit řízení pomocí logiky, viz Obr. 35. Virtuální zprovoznění v *event-based* režimu lze několika způsoby:

❖ Cyclic Event Evaluator (CEE)

Je řízení událostmi. Pořadí operací se řídí pomocí logiky opírající se o aktuální stav jednotlivých signálů. Funguje na principu řídicího centra, kdy během každého cyklu shromáždí a vyhodnotí PLC signály a určí následující dění v simulaci [10]. Samotné PLC tudíž nahrazuje přímo PS a logické řídicí instrukce se definují jako tzv. *Modules*.

❖ SIMIT

Slouží pro přímé napojení na simulační nástroj SIMIT, kde následně umožňuje propojit PLC prostřednictvím Profinet, Profibus nebo OPC serveru. Simulace může být tak řízena reálným PLC [19].

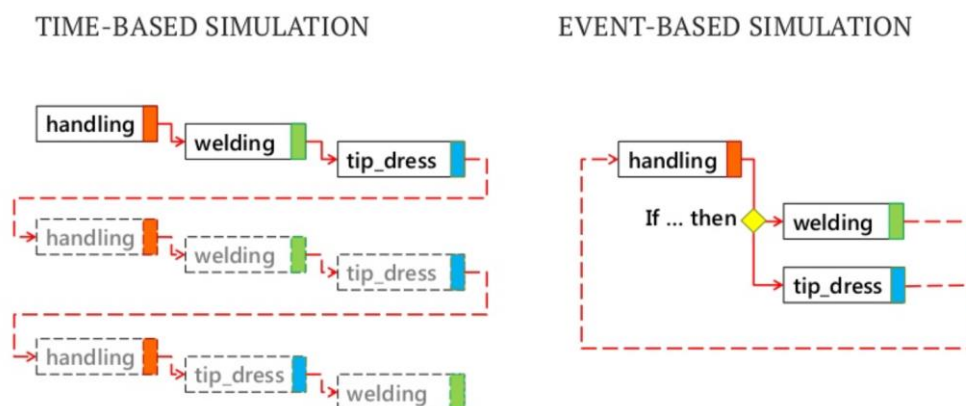
❖ PLCSIM

Rozšířená verze STEP 7 Professional, jež je základní programovací nástroj pro PLC systémy SIMATIC, obsahuje modul SIMATIC S7–PLCSIM. Modul slouží pro vytvoření virtuálního PLC, jež lze pomocí možnosti PLCSIM propojit s PS [19].

❖ External Connection

Univerzální způsob pro propojení PS se softwarem nebo zařízením třetích stran. Předpoklad úspěšného připojení je, že existuje již vytvořený OPC server a PS se připojí pouze jako klient. Na druhé straně lze připojit teoreticky jakékoliv zařízení nebo software podporující OPC komunikační standard.

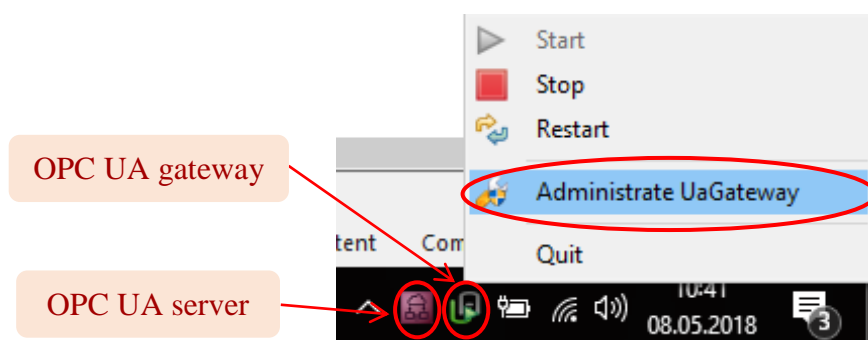
V diplomové práci byla pro virtuální zprovoznění použita koncepce SoftPLC od firmy Beckhoff, jež se programuje v programu TwinCAT. Tato koncepce je pro dané pracoviště nejvhodnější, neboť reálné pracoviště bude obsahovat řídicí systém Beckhoff včetně OP Beckhoff. Jelikož softwary jsou od různých společností, vzniká drobná komplikace v jejich vzájemném propojení. Proto je nutné použít univerzální komunikační protokol OPC UA jako prostředníka k realizaci funkčního propojení. Podmínkou je pouze podpora zmíněného protokolu v obou softwarech, což splňují.



Obr. 35) Rozdíl ve vykonání stejného procesu ve dvou režimech PS [1]

7.1 Nastavení OPC serveru

Prvním krokem k propojení je instalace OPC serveru. K dispozici je více než desítka instancí fungujících na podobné bázi. Vzhledem k tomu, že provádím veškeré řízení na PLC firmy Beckhoff, volím OPC instanci TF6100 TC3 OPC UA, což je doplněk programu TwinCAT. Samozřejmostí je nejdříve instalace samotného programu a posléze zmíněného doplňku. Kompletní instalace doplňku obsahuje OPC UA server a UA Gateway, viz Obrázek 36. Konfigurace serveru není nutná, jelikož samotná instalace doplňku vytvoří vhodnou strukturu s možností připojení jednoho PLC.



Obr. 36) Prvky OPC UA instance v tray liště

Nicméně *Gateway* je vhodné překontrolovat v možnosti *Administrate UaGateway* (Obr. 36). V kartě *General* je vhodné mít zapnuté funkce *UaGateway Runtime Process* pro automatické spuštění při startu počítače a *Notification Area Icon* pro zobrazení *Gateway* v systémové liště k jednoduššímu přístupu. Skupina *Configuration Permissions* uděluje oprávnění pro správu *Gateway*, aby neautorizovaní uživatelé nemohli odstranit připojení ke konkrétnímu serveru nebo přemostit připojení na server jiný. Pro účely testování doporučuji ponechat možnost *Everyone*.

V kartě *Configured UA Servers* se ponechá výchozí nastavení (Obr. 37). *Server Name* je volitelné. *Server Url* žádá cestu k OPC serveru. Skládá se z předdefinovaného „opc“, volby komunikačního protokolu (TCP, HTTPS, HTTP), IP adresy (například localhost, 192.168.0.100) a portu (4840, 4850) [18]. Pokud se OPC server nachází na stejném počítači, jako je *Gateway*, ponechá se výchozí nastavení. Jestliže se server nachází na jiném zařízení, je nutné zjistit jeho IP adresu a zkontrolovat průchod zvoleného protokolu v síťových periferiích. Situace může nastat, pokud softwary určené ke vzájemnému propojení jsou nainstalovány na různých počítačích. V takovém případě je nutno doinstalovat *Gateway* (nikoliv OPC server) i na druhé zařízení, kde se nenachází OPC server, a řádně ho nakonfigurovat. Další položky obstarávají zabezpečení, resp. jsou-li vyžadovány přihlašovací údaje ze strany serveru, zde se zadají. Pro účely testování postačí anonymní nezabezpečené připojení, jelikož tuto možnost OPC server povoluje. Poslední položka prezentuje, zdali má konfigurace být aktivní.

Karta *UA Endpoints* diktuje, jakým způsobem se mohou klienti přes *Gateway* připojit k OPC serveru. *Endpoints* se definují pro klienty nacházející se na stejném zařízení jako *Gateway*. Doporučuji ponechat výchozí nastavení automaticky vygenerované po nakonfigurování OPC serveru.

Configured UA Servers

Server Name	Server Url	Security	User Token Type	Connect to Server
TcOpcUaServer	opc.tcp://localhost:4840		Anonymous	true

Obr. 37) Výchozí nastavení připojení k OPC serveru

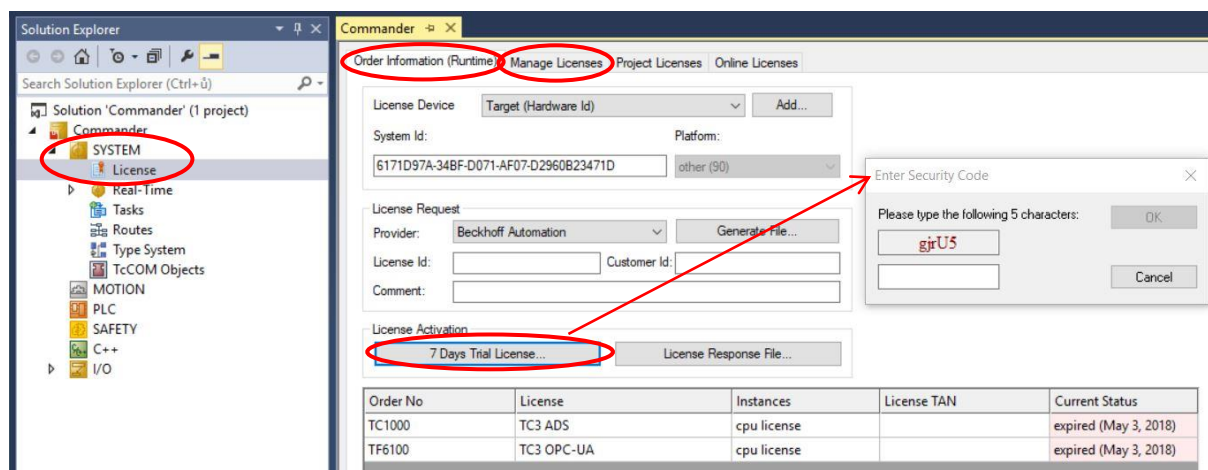
7.2 Nastavení SW TwinCAT

Pro vytvoření řídicího PLC jsem zvolil nejnovější verzi programu TwinCAT 3 běžícího na jádře Microsoft Visual Studio 2013. Při prvním spuštění na platformě Windows 8 (a výše) je nutné nejdříve spustit konfigurační dávkový soubor *C:\TwinCAT\3.1\System\win8settick.bat*. Díky tomu může program spustit PLC, jež běží v reálném čase. Respektive provede veškeré příkazy ve stejném časovém cyklu (např. 10 ms). Nový cyklus započne až po uplynutí zmíněné doby. Výchozí nastavení Windows verze 8 a výše takového chování programu nedovoluje.



Obr. 38) Spuštění SW TwinCAT

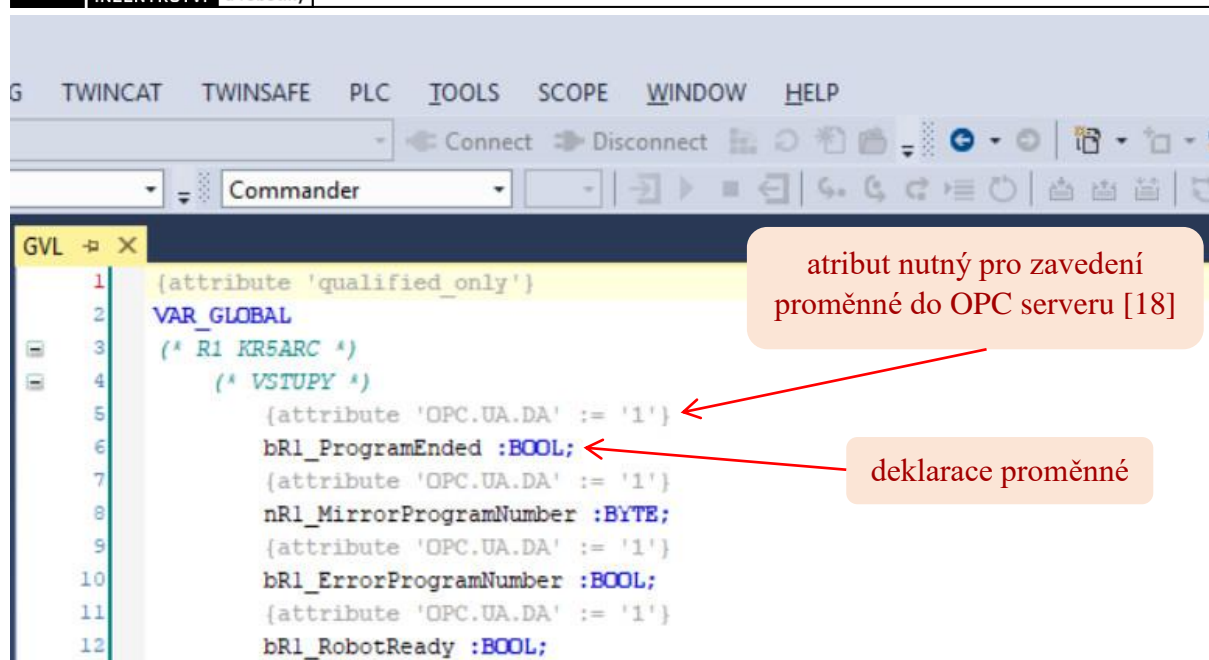
Po spuštění programu TwinCAT (Obr. 38) je třeba založit plnohodnotný projekt *FILE* → *New* → *Project*. V dialogovém okně se vybere ze šablon *TwinCAT Projects* a ve střední části okna se zvolí položka *TwinCAT XAE Project*. Je vhodné si projekt před založením řádně pojmenovat a zvolit umístění. Pokud se jedná o studentskou verzi programu TwinCAT, nejdříve se vygenerují dočasné licence na samotný program a doplněk OPC–UA. V levém okně *Solution Explorer* se rozbalí skupina *SYSTEM* a zvolí položka *License*. V pracovní oblasti programu se objeví dialogové okno se zadaným názvem celého projektu. V kartě *Manage Licenses* je nutné najít doplněk s názvem *TF6100* a zaškrtnout položku *cpu license* ve sloupci *Add License*. Nyní se v první kartě *Order Information (Runtime)* vygenerují sedmidenní licence pomocí tlačítka *7 Days Trial License*, viz Obrázek 39.



Obr. 39 Prodloužení licencí v SW TwinCAT

Dalším krokem je vytvoření PLC instance pomocí pravého kliknutí na skupinu *PLC* → *Add New Item*. Použije se šablona *Standard PLC Project*. Není vhodné opomenout na řádné pojmenování. Umístění doporučuji ponechat výchozí, jelikož se vytvořená PLC instance automaticky uloží do složky nadřazeného projektu. Důležité je zaškrtnout položku *TMC File* umístěná v kartě *Settings* po kliknutí na nově vytvořenou PLC instanci, jež se objeví pod volitelným názvem ve skupině *PLC*.

Nyní lze přistoupit k deklaraci proměnných. Jelikož OPC Unified Architecture podporuje propojení proměnných ze dvou softwarů různých výrobců pouze pomocí jejich názvů, není potřeba složité adresace. Postačí pouze přesně přepsat názvy signálů z PS do programu TwinCAT. Ve většině případů je velký počet sdílených vstupních a výstupních signálů, proto by bylo náročné ruční přepsání. Lze použít postup, jenž využívá funkci programu PS. Má název *Export to Excel* a lze ji nalézt v *View* → *Viewers* → *Signal Viewer*. Po otevření vytvořeného souboru se zobrazí veškeré signály a pomocí funkce *Concatenate* v programu Excel se snadno vytvoří formát požadovaný programem TwinCAT pro řádnou deklaraci. Veškeré proměnné se vloží do souboru *GVL*. Rozbalením PLC instance se zobrazí složka *GVLs*, do které se vloží soubor se zmíněným názvem *GVL* pravým kliknutím na *GVLs* → *Add* → *Global Variable List*. Obrázek 40 prezentuje část souboru *GVL* pro snazší pochopení specifické deklarace.

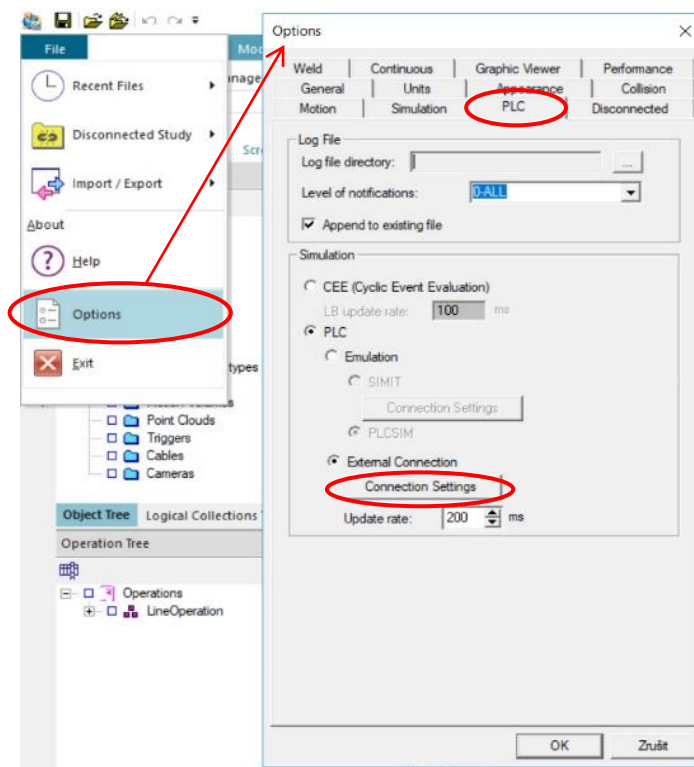


Obr. 40) Ukázka deklarace proměnných v souboru GVL

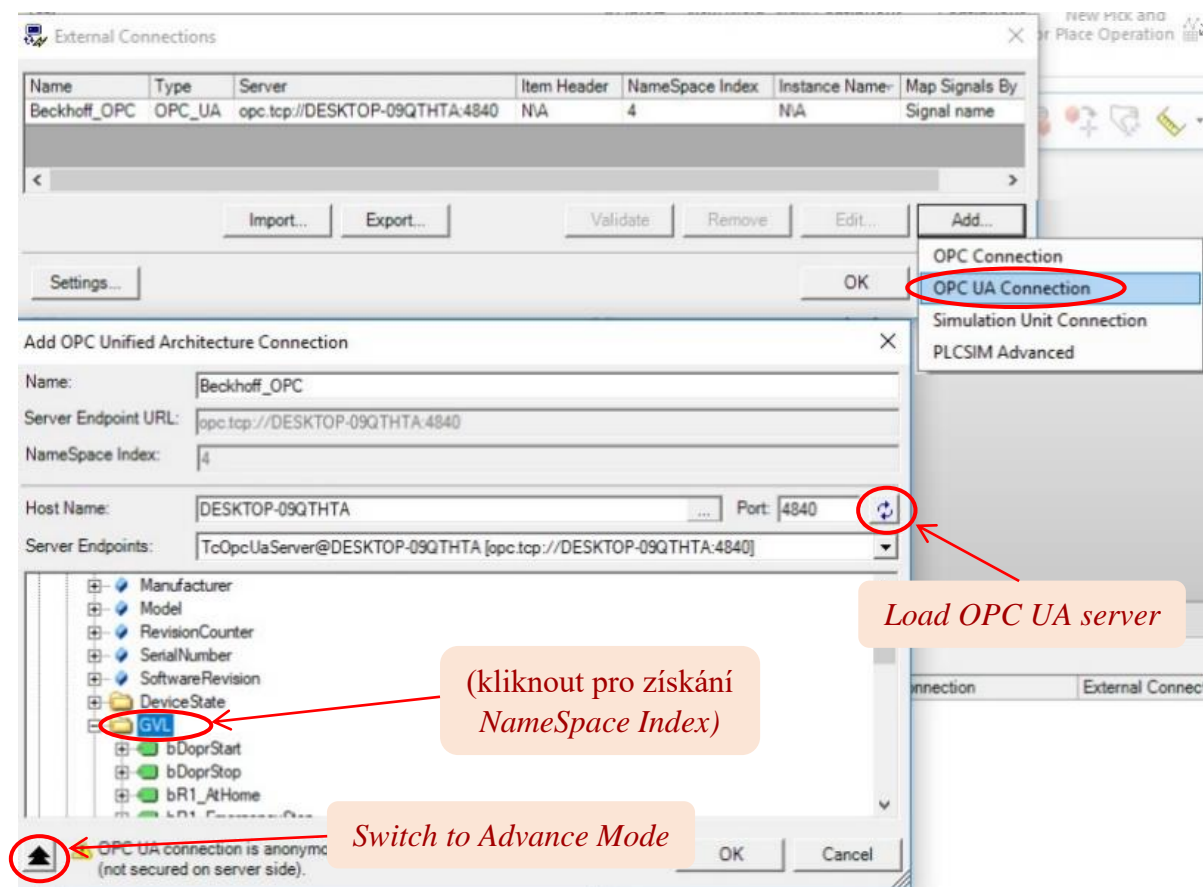
7.3 Nastavení SW PS

Aby mohl PS řádně komunikovat, je důležité spustit program jako správce. V platformě Windows lze tuto možnost aktivovat permanentně pravým kliknutím na spouštěcí ikonu → *Vlastnosti* → *Kompatibilita* → *Nastavení* → *Spustit tento program jako správce*. Při běžném spuštění by program nenalezl vytvořený OPC server. Následující nastavení platí pro veškeré projekty včetně nově vytvořených.

Po otevření dialogového okna *File* → *Options* lze v kartě *PLC* nastavit externí připojení k OPC serveru, viz Obrázek 41. Nové připojení se přidá tlačítkem *Add* → *OPC UA Connection*. Otevře se nové dialogové okno. Položka *Name* je volitelná. Další dvě zbývající položky žádají URL adresu *Endpointu* (včetně komunikačního protokolu a portu) dříve vytvořeného v *Gateway*. Jelikož URL adresa nespecifikuje přesnou lokaci v adresářové struktuře OPC serveru, je nutné proměnné lokalizovat pomocí *Namespace Index*. Pokud jsou tyto údaje neznámé, lze použít automatické vyhledání pomocí tlačítka *Switch to Advance Mode* (obrázek černé šipky). Po kliknutí se rozšíří dialogové okno (Obr. 42) a automaticky se program připojí k OPC serveru na lokálním zařízení (pokud existuje). V případě, že se OPC server nachází na jiném zařízení, je nutné zadat do kolonky *Host Name* IP adresu cílového zařízení a kliknout na tlačítko *Load OPC UA server content* (modré šipky). Pokud je připojení k OPC serveru funkční, zobrazí se jeho adresářová struktura. Nyní postačí vyhledat proměnné předem vytvořené v programu TwinCAT a dvakrát poklepat na jejich nadřazenou složku. V mém případě se jedná o složku *Objects/PLC1/GVL*. Díky tomuto úkonu se změní hodnota *Namespace Index* z nuly na jiné přirozené číslo. Následně se vše potvrdí tlačítkem *OK*. Před pozdějším zahájením simulace doporučuji přezkontrolovat funkčnost propojení tlačítkem *Validate* v dialogovém okně *External Connections*.



Obr. 41) Karta *PLC* v dialogovém okně *Options*



Obr. 42) Dialogové okno *External Connections*

Druhá generace OPC Unified Architecture přináší řadu výhod, nichž je nejdůležitější (v konkrétním případě) propojení jednotlivých signálů pomocí jejich názvů. Proto se stačí zaměřit pouze na názvy signálu a není třeba řešit složitou adresaci. Nicméně je nutné signálům:

- ✓ přiřadit do jejich názvů prefix „GVL.“ podle názvu souboru, ve kterém jsou proměnné v programu TwinCAT deklarovány,
- ✓ zvolit, zda má signál komunikovat s PLC,
- ✓ zvolit předem vytvořené externí připojení k OPC serveru (v jiných případech může být projekt řízen pomocí několika OPC serverů zároveň).

Po otevření *View* → *Screen Layout* → *Viewers* → *Signal Viewer* se v okně vypíší veškeré signály projektu. Ve většině případů by bylo náročné měnit atributy pro každý signál manuálně, proto se s výhodou využijí funkce *Export to Excel* (pro export tabulky) a *Signal Mapping Tool* (pro import signálu s nastavenými atributy), jež jsou součástí okna *Signal Viewer*.

Nejprve je nutné vyexportovanou tabulku převést do sloupců, jelikož jsou sloupce umístěny v jedné buňce oddělené oddělovačem – čárkou. Postup je obdobný jako při vytváření proměnných pro OPC server v programu TwinCAT, jenž je popsán v kapitole 7.2. Nejdříve se ve sloupci *PLC Connection* vypíše hodnota *True* pro každý signál, jenž je požadován externí automatikou, resp. OPC serverem. U zbylých se vypíše *False*. Následně se řádky seřadí podle tohoto sloupce sestupně. Signály s atributem *True* budou pohromadě nahoře a po nich signály s atributem *False*. Pro horní část signálu (s atributem *True*) se do sloupce *External Connection* vypíše volitelný název pro předem vytvořené připojení, jež se provedlo v dialogovém okně *Options* v programu PS. V mém případě se jedná o název *Beckhoff_OPC*. V neposlední řadě je potřeba vytvořit nový sloupec mezi sloupci *Signal Name* a *Memory* s názvem *Electrical Name*. Atribut slouží pro přiřazení nového názvu ke stávajícím signálům v programu PS. Zde se uvede zmíněný prefix „GVL.“ s původním názvem. Opět pouze pro signály určené ke sdílení s OPC serverem. Pro rychlejší zadání se využije funkce *Concatenate* programu Excel. Obrázek 43 částečně zachycuje již připravený dokument k importu se signály zpět do programu PS pomocí dříve zmíněné funkce *Signal Mapping Tool*. Upravený dokument je nutné uložit s příponou *.xls nebo *.xlsx, neboť funkce *Export to Excel* vytvořila dokument s příponou *.CSV.

SUMA										
=CONCATENATE("GVL";A2)										
	A	B	C	D	E	F	G	H	I	J
	Signal Name	Electrical Name	Memory	Type	Address	IEC Format	PLC Connection	External Connection	Resource	Comment
1	bDoprStart	=CONCATENATE("GVL";A2)	False	BOOL	No Address	No Address	True	Beckhoff_OPC	DopravnikPasovy	
3	bDoprStop	GVLbDoprStop	False	BOOL	No Address	No Address	True	Beckhoff_OPC	DopravnikPasovy	
4	br2_StartProgram	GVLbr2_StartProgram	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
5	nR2_ProgramNumber	GVLnR2_ProgramNumber	False	BYTE	No Address	No Address	True	Beckhoff_OPC	kr16_2	
6	br2_EmergencyStop	GVLbr2_EmergencyStop	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
7	br2_ProgramPause	GVLbr2_ProgramPause	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
8	br2_ProgramEnded	GVLbr2_ProgramEnded	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
9	nR2_MirrorProgramNumber	GVLnR2_MirrorProgramNumber	False	BYTE	No Address	No Address	True	Beckhoff_OPC	kr16_2	
10	br2_ErrorProgramNumber	GVLbr2_ErrorProgramNumber	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
11	br2_RobotReady	GVLbr2_RobotReady	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
12	br2_AtHome	GVLbr2_AtHome	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr16_2	
13	br1_StartProgram	GVLbr1_StartProgram	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr5_arc	
14	nR1_ProgramNumber	GVLnR1_ProgramNumber	False	BYTE	No Address	No Address	True	Beckhoff_OPC	kr5_arc	
15	br1_EmergencyStop	GVLbr1_EmergencyStop	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr5_arc	
16	br1_ProgramPause	GVLbr1_ProgramPause	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr5_arc	
17	br1_ProgramEnded	GVLbr1_ProgramEnded	False	BOOL	No Address	No Address	True	Beckhoff_OPC	kr5_arc	

Obr. 43) Signály připravené na import

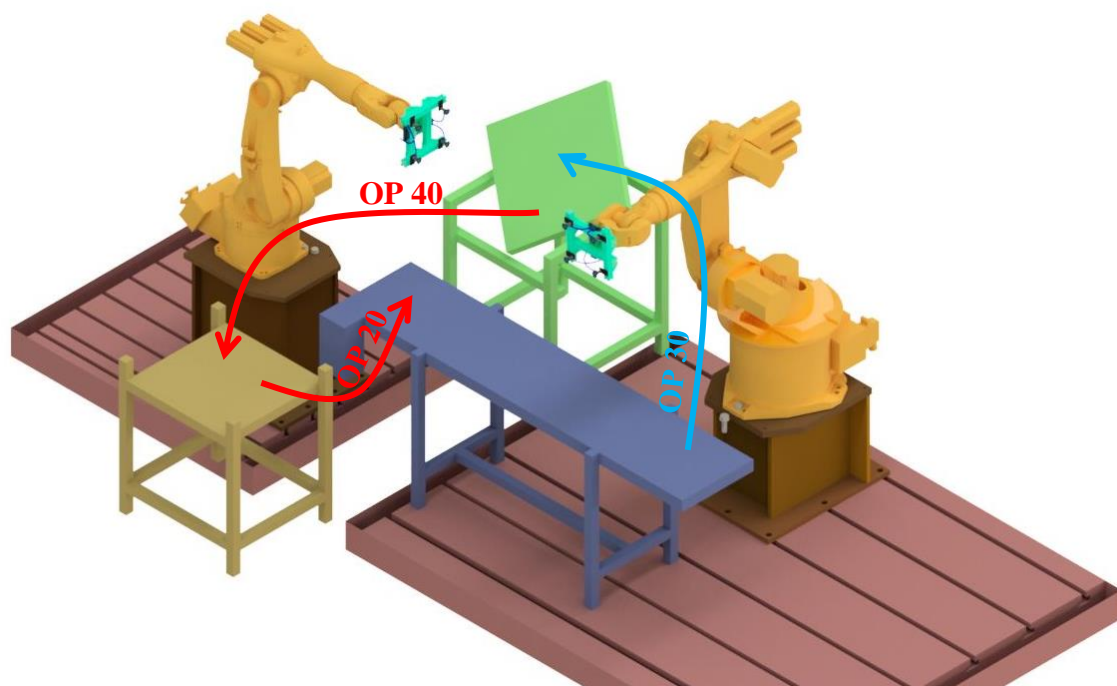
7.4 Logika řízení robotického pracoviště

K zadanému projektu je potřeba vypracovat funkční logické řízení externí automatiky. Následující podkapitoly popisují řídicí algoritmy a popis jednotlivých dílčích periférií pracoviště a operací.

7.4.1 Robotické operace

Robot R1 v zadaném projektu disponuje dvěma operacemi pro manipulační činnost, jednou operací pro servisní práce a jednou operací pro inicializaci. Operace číslo 20 má za úkol přemístit součást ze vstupní části stolu (místo pro zakládání součástí) do vstupní části dopravníku, kde následné přemístění realizuje dopravní pás. Druhá operace, číslo 40, plní funkci paletizační. Nejprve se součást vycentruje pomocí tíhové síly na skluzu. Následně umožní robotu R1 součást uchopit v předem známé pozici, tj. TCP na středu povrchu součásti. To umožní přesnou paletizaci do stohu na výstupní části stolu. Třetí operace, číslo 110, slouží pro přemístění robota R1 do předem stanovené pozice. Výsledná poloha usnadní obsluhu případnou údržbu ejektoru, přísavek nebo celého efektoru. Poslední operace, číslo 120, slouží k přemístění robota R1 do polohy *Home*, čímž dojde k přepnutí signálu *bR1_RobotReady* do stavu *True*. Spuštění proběhne vždy při novém startu automatického cyklu. Avšak operace 120 slouží primárně k aktivaci zmíněného signálu *bR1_RobotReady*, jelikož najetí do polohy *Home* z neznámé pozice není přípustné z důvodu možné kolize. Tudíž je nutné před automatickým cyklem ručně přemístit robota do pozice *Home*.

Pro robota R2 jsou vytvořeny celkem tři operace. Jakmile dopravník přemístí součást na konec, spustí se operace 40. Robot R2 uchopí součást, kdy TCP se nachází v přibližném středu na povrchu součásti. Přepokládá se pouze malá odchylka (do 5 mm) TCP od středu součásti. Následně je součást přemístěna do skluzu tak, aby mohla sklouznout do vycentrované polohy. Tudíž se počítá s mírně nepřesným uchopením součásti efektozem. Zbylé dvě operace jsou totožné s operacemi robota R1. Operace číslo 120 plní stejnou funkci jako operace číslo. Pro inicializační přesunutí robota R2 do *Home* pozice slouží operace číslo 125.



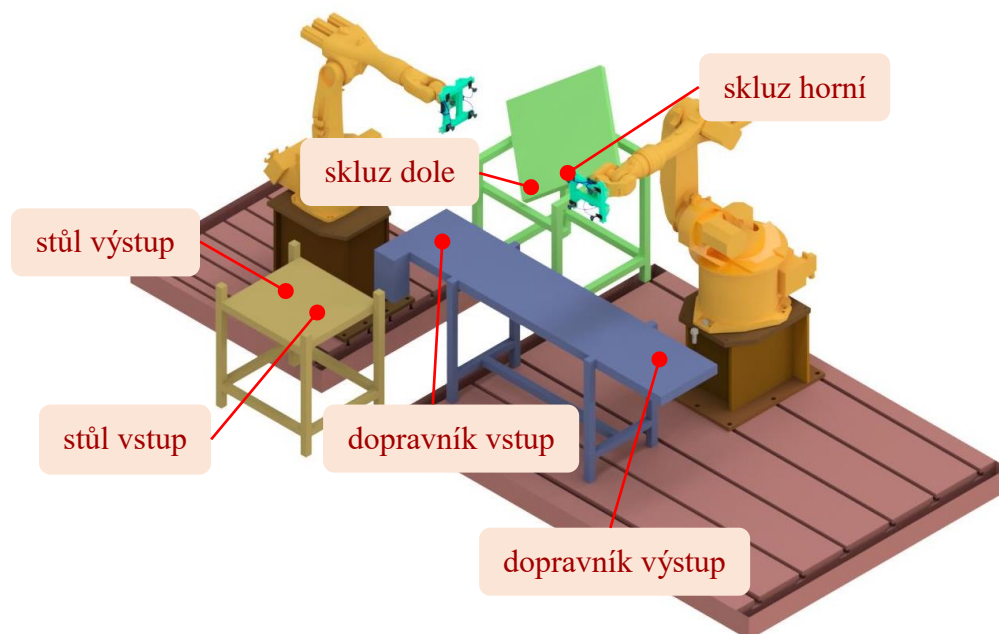
Obr. 44) Znázornění operací v pracovišti

7.4.2 Senzory

Pro správnou funkčnost automatického cyklu jsem použil celkem šest senzorů hlídající pouze přítomnost součástí. Spuštění a zastavení dopravníku hlídají dva senzory, jež jsou umístěny na začátku a konci pásu. Jakmile součást dojde na konec dopravníku, automatika zastaví pás pouze pomocí čidla signalizujícího přítomnost kusu. Tudiž v zadaném projektu není potřeba, aby jakýkoliv senzor hlídal přiblížení součásti. Postačí pouze, aby senzor odesílal informaci *True* (součást je přítomna na daném místě) a *False* (na daném místě se nenachází žádná součást). Obrázek 45 prezentuje umístění senzorů v zadaném projektu.

Na přípravku – stůl se nacházejí dva senzory. Vstupní hlídá, zdali stále existuje součást určená k odbavení. Jakmile dojde k odbavení všech součástí ze vstupního stolu, senzor předá informaci externí automaticce, jež zamezí robotu R1 vykonat operaci číslo 20. Ve virtuální simulaci navíc předá informaci, kdy je potřeba vygenerovat novou část. Výstupní senzor je účelný zejména pro inicializační část. Určí, zdali je výstupní místo prázdné, aby mohl robot R1 začít paletizovat novou dávku. Ve virtuální simulaci slouží k informaci, kdy součást dorazila na konec cyklu a je ji potřeba odstranit.

Na přípravku – skluz jsou záměrně umístěny dva senzory, i když k funkčnosti by stačil pouze senzor jeden hlídající přítomnost kusu. Faktem je, že informace o přítomnosti kusu se dá vyčíst i z ejektoru VADMI. Pokud se nachází uvnitř přísavky vakuum při vykládání součásti na skluz, je zaručeno, že součást během pohybu robota R2 dorazila na cílové místo. Nicméně není zaručeno, že součást vlivem tíhové síly se přemístí správně orientována na místo určené k uchopení robotem R1 v operaci 40. Tento scénář je sice nepravděpodobný, ale pokud by se stal, mohlo by dojít k poškození efektoru a automatický cyklus by tuto chybu nezaznamenal. Proto se nachází na skluzu dva senzory s roztečí šířky součásti, tj. 250 mm. Jakmile senzory vyšlou rozdílnou informaci o přítomnosti kusu, vznikne podezření na špatně založenou (resp. orientovanou) součást na skluzu.



Obr. 45) Umístění senzorů na pracovišti

7.4.3 Signály

Tabulka níže vysvětluje veškeré signály, jež obsahuje OPC server. Jedná se o komunikační cesty, pomocí kterých nadřazené PLC řídí celé pracoviště. Tabulka nezachycuje signály pro robota R2, jelikož jsou totožné se signály pro robota R1.

Tab 4) Signály pro komunikaci PS s PLC

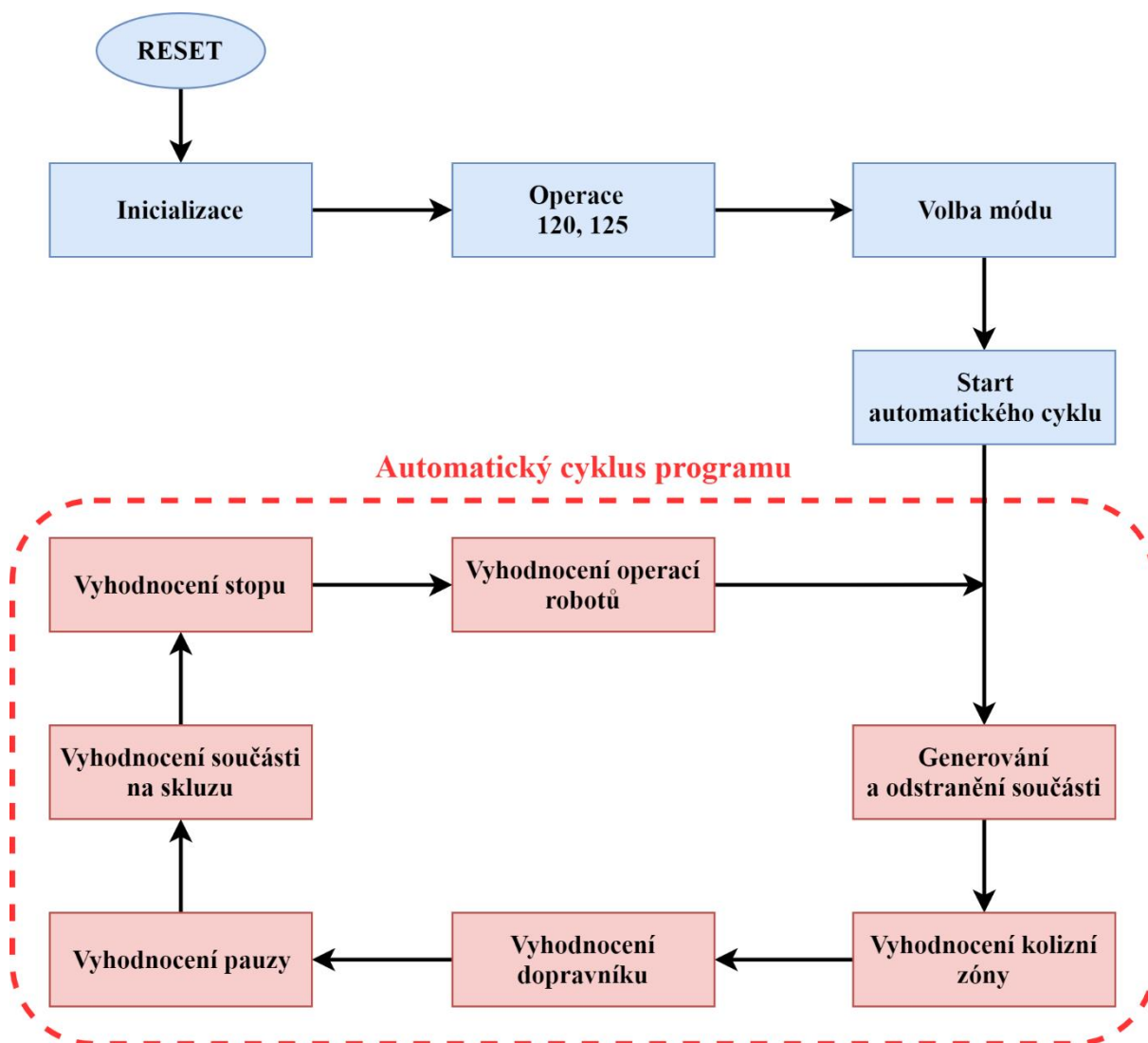
Název signálu	Typ	I/O	Popis
Robot R1			
bR1_StartProgram	BOOL	O	start aktuálně zadané operace; reakce na RE ¹
nR1_ProgramNumber	BYTE	O	číslo operace pro vykonání robotem
bR1_EmergencyStop	BOOL	O	nouzové zastavení všech pohybů robota
bR1_ProgramPause	BOOL	O	pozastavení aktuálně probíhající operace; RE pro pozastavení a FE ² pro opětovné spuštění
bR1_ProgramEnded	BOOL	I	oznámení o ukončení operace; hodnota <i>True</i> dokud se nepustí operace nová
nR1_MirrorProgramNumber	BYTE	O	v zadaném projektu nepoužitý signál
bR1_ErrorProgramNumber	BOOL	I	chybě zadané číslo operace; operace se zadaným číslem neexistuje
bR1_AtHome	BOOL	I	robot v <i>Home</i> pozici
bR1_RobotReady	BOOL	I	robot ukončil operaci a je připraven na další
bR1_DoprFree	BOOL	O	prázdné vstupní místo na dopravníku
bR1_NaSkruz	BOOL	I	robot se právě nachází v kolizní zóně.
bR1_ZoneFree	BOOL	O	zaručen robotu bezpečný vstup do kolizní zóny
Dopravník			
bDoprStart	BOOL	O	start dopravníku; reakce pouze na RE
bDoprStop	BOOL	O	stop dopravníku; reakce pouze na RE
nDoprSpeed	INT	O	rychlost dopravníku
Generování součástí			
bOutPart	BOOL	O	odstranění součásti z výstupního místa stolu; reakce pouze na RE
bGenPart	BOOL	O	vytvoření součásti na vstupní části stolu; reakce pouze na RE
Senzory			
bSensorSkruzDole	BOOL	I	dolní senzor na skluzu
bSensorSkruzHorni	BOOL	I	horní senzor na skluzu
bSensorDoprVstup	BOOL	I	senzor na vstupním místě dopravníku
bSensorDoprVystup	BOOL	I	senzor na výstupním místě dopravníku
bSensorStulVystup	BOOL	I	senzor na vstupní části stolu
bSensorStulVstup	BOOL	I	senzor na výstupní části stolu

¹ RE = rising edge neboli náběžná hrana signálu

² FE = falling edge neboli sestupná hrana signálu

7.4.4 Procesy hlavního PLC programu

Řízení celého pracoviště lze rozdělit do jedenácti hlavních procesních skupin, viz Obrázek 46. Počínaje resetem PLC program provede zaváděcí procesy potřebné pro řádný chod automatického cyklu. V případě, že se pracoviště nenachází v žádaném stavu, informuje obsluhu. Jakmile operátor vyřeší veškeré nesrovnalosti, stiskne tlačítko *RESET* a zaváděcí procesy znovu započnou s kontrolou stavu robotické buňky pro možné spuštění automatického cyklu.



Obr. 46) Hlavní skupiny procesů PLC programu

➤ Inicializace

Při novém startu řídicí jednotky, je nutné definovat výchozí stav všech proměnných a periférií v robotické buňce. Proces *Inicializace* pomáhá identifikovat stav vstupních proměnných (z pohledu PLC) a nastavit inicializační hodnoty pro proměnné výstupní. Díky čemuž lze zjistit aktuální stav pracoviště.

➤ Operace 120, 125

Provádí spuštění inicializačních operací pro roboty R1 a R2, viz kapitola 7.4.1. Kontrola, zdali se roboty nacházejí v *Home* pozici, se provádí v inicializačním procesu.

➤ Volba módu

Slouží pro přepnutí mezi dvěma módy. Servisní mód slouží k údržbě, diagnostice a testování funkčnosti dílčích periférií. Normální mód obsahuje automaticky cyklus pro manipulační operace v robotické buňce.

➤ Start automatického cyklu

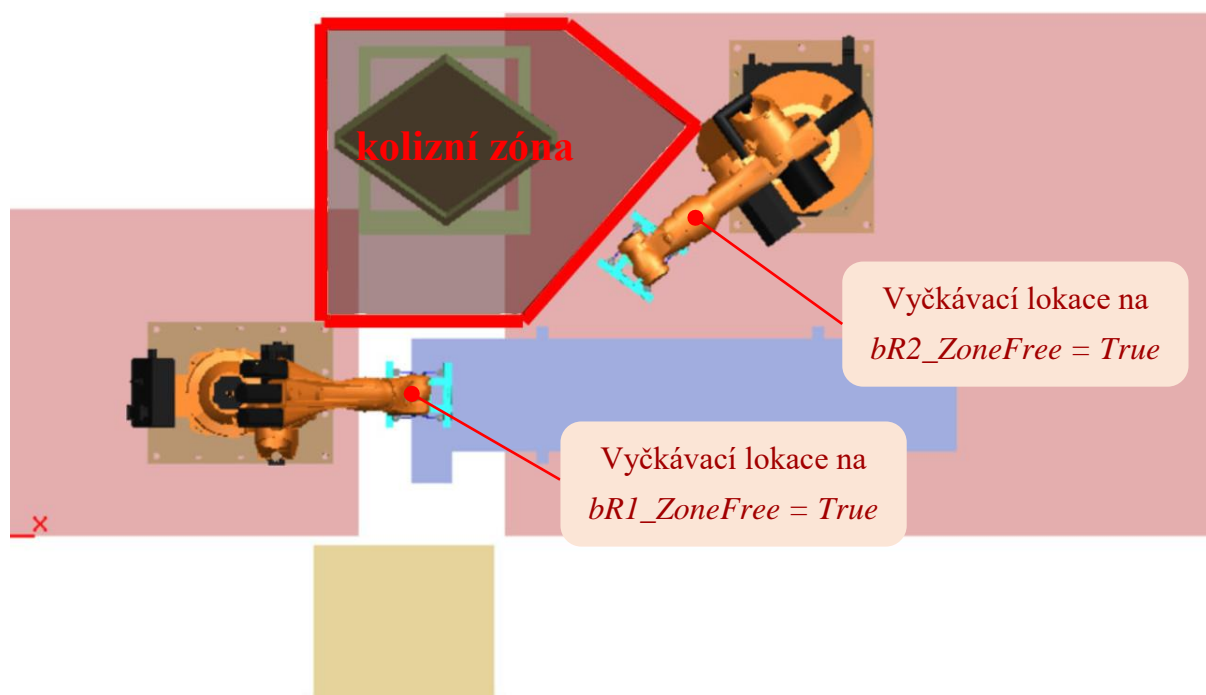
Proces sloužící k vyhodnocení startu, pauzy a stopu.

➤ Generování a odstranění součástí.

Pomocí dvou výstupních (z pohledu PLC) signálů proces řídí generování součástí ze vstupního místa pracoviště a odvod součástí z výstupního místa. Je vytvořen tak, aby automatický cyklus fungoval v nekonečné smyčce.

➤ Vyhodnocení kolizní zóny

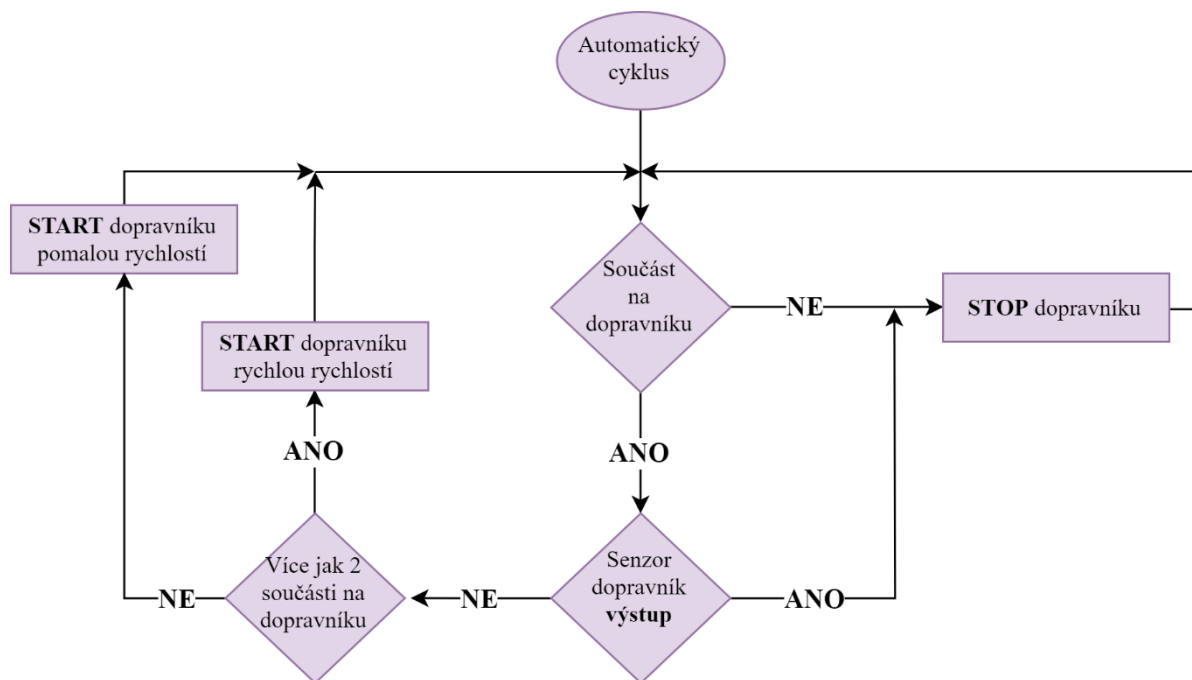
Jedná se o zónu poblíž skluzu (Obr. 47), kde mohou roboty přijít do vzájemné kolize a způsobit hmotné škody. Hlavní funkci procesu je povolení jednotlivým robotům vstup do zmíněné zóny pomocí vyhodnocení vstupních signálů. Sekundární funkci je vyhodnocení přítomnosti součástí na skluzu. Proces předává informaci robotům pomocí výstupního signálu s názvem *ZoneFree*.



Obr. 47) Kolizní zóna

➤ Vyhodnocení dopravníku

Pod uvedeným pojmem se skrývá veškeré řízení dopravníku. Jedná se o spuštění pohonu, zastavení a změnu rychlosti. Reverzace dopravníku v zadaném projektu není možná. Obrázek 48 vyobrazuje rozhodovací algoritmus.



Obr. 48) Flowchart k řízení dopravníku

➤ Vyhodnocení pauzy

Po stisknutí tlačítka pauzy proces podnikne nutné kroky k pozastavení činnosti v robotickém pracovišti. Roboty i dopravník jsou ihned pozastaveny. Jakmile takto učiní, předá další činnost procesu *Start automatického cyklu*, kde se může provést pokračování automatického cyklu.

➤ Vyhodnocení součásti na skluzu

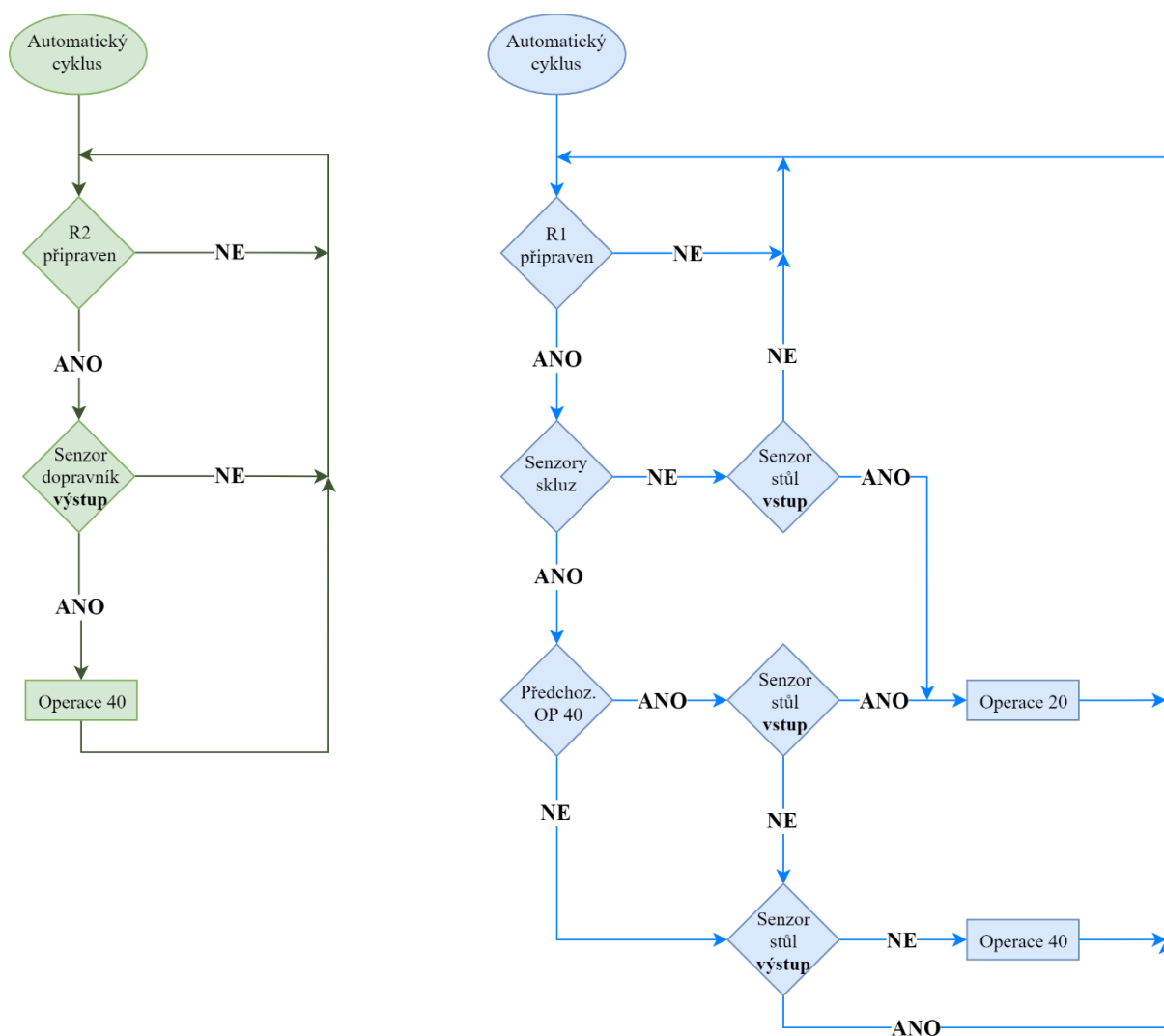
Pokud oba senzory na skluzu ukazují totéž, tj. stejnou informaci *True* nebo *False*, proces se nespustí a přeskočí se. Jakmile obsahují signály rozdílnou informaci, spustí se časovač nastavený na tři sekundy. Čeká se, jestli se situace sama nenapraví, tj. součást se přesune do vycentrované pozice. Pokud rozdílná informace trvá dále, automatický cyklus je zastaven a vyvěsí příslušnou chybu na uživatelské rozhraní.

➤ Vyhodnocení stop

Pokud obsluha zastaví automatický cyklus tlačítkem stop, proces nedovolí spuštění nové operace pro roboty. Všechny aktuálně běžící operace se nechají dokončit. Dopravník je zastaven (pokud je v chodu) jakmile dopraví součást do výstupní pozice dopravníku.

➤ Vyhodnocení operací robotů

Jedná se o nejdůležitější proces v celém PLC programu, jelikož přikazuje oběma robotům, jakou operaci mají právě vykonat. Obrázek 49 slouží k snadnějšímu pochopení algoritmu. Zmíněný rozhodovací algoritmus je důležitý pro správný chod robotické buňky. Pokud by se vybrala nesprávná následnost operací, mohlo by dojít k nepříznivému prodloužení taktu celého pracoviště. Jelikož se v zadaném projektu jedná o virtuální zprovoznění, lze díky testovacím simulacím docílit optimalizované logiky k řízení operací pro roboty.



Obr. 49) Flowchart k řízení robotů
vlevo — řízení robota R2 (KR 16-2); vpravo — řízení robota R1 (KR 5 ARC)

7.4.5 Vizualizace

Jelikož každé robotické pracoviště potřebuje být nějakým způsobem ovládané, je nasnadě vytvořit ovládací panel jako HMI (human machine interface). K tomuto účelu jsem vytvořil dvě vizualizace sloužící jako uživatelské prostředí pro ovládání robotické buňky pomocí dotykové obrazovky řídicího, resp. operátorského panelu, což je běžný způsob používání i v průmyslové praxi. Vizualizace jsou přizpůsobeny pro 15" monitor s rozlišením 1024 x 768 obrazových bodů.

❖ Normální mód

Na Obrázku 50 lze pozorovat základní obrazovku, jež klasifikuji jako tzv. normální mód. Používá se k běžnému ovládání robotické buňky. Operátor díky ní může aktivovat automatický cyklus. Stačí aby se řídil podle pokynů, jež se zobrazují v kolonce *Aktuální stav*. Obrazovka slouží také jako informační panel. Informační prvek je:

- Uběhlý čas — celkový čas od prvního založeného dílu do pracoviště (lze resetovat),
- Takt pracoviště — průměrná doba trvání přesunu součásti ze vstupního místa stolu do místa výstupního (lze resetovat),
- Počet plechů na páse — aktuální počet součástí na páse dopravníku,
- Počet plechů celkem — celkový počet odbavených plechů (lze resetovat),
- Aktuální stav — informace o průběhu pracoviště včetně pokynů pro operátora,
- Chybový stav — popis vzniklé chyby na pracovišti.

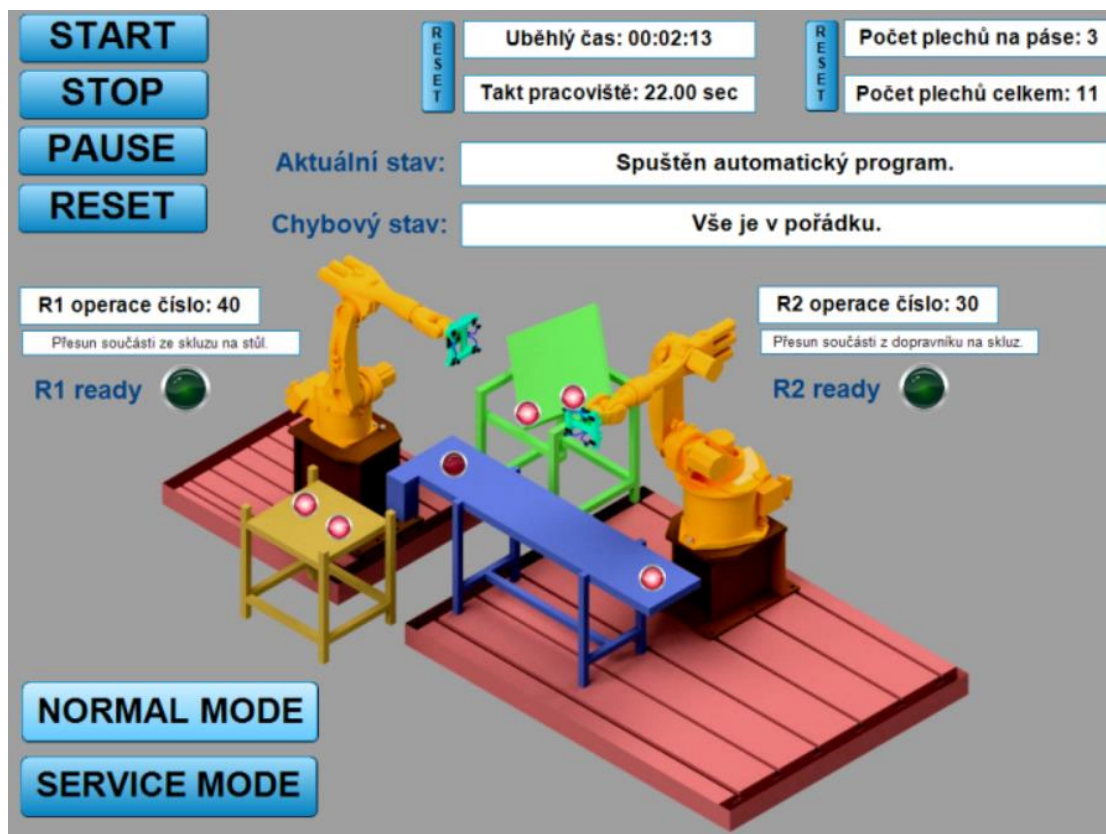
Na vizualizaci lze pozorovat i další informační prvky, jež v reálném čase informují o aktuálním stavu obou robotů a všech klíčových senzorů. Pro ovládání slouží sada tlačítek s funkcí:

- START — spuštění automatického cyklu, zrušení pauzy,
- STOP — blokáce spuštění všech nových operací,
- RESET — kompletní nová inicializace celé robotické buňky,
- NORMAL MODE — přepnutí na základní obrazovku (Obr. 50),
- SERVICE MODE — přepnutí na servisní obrazovku (Obr. 51).

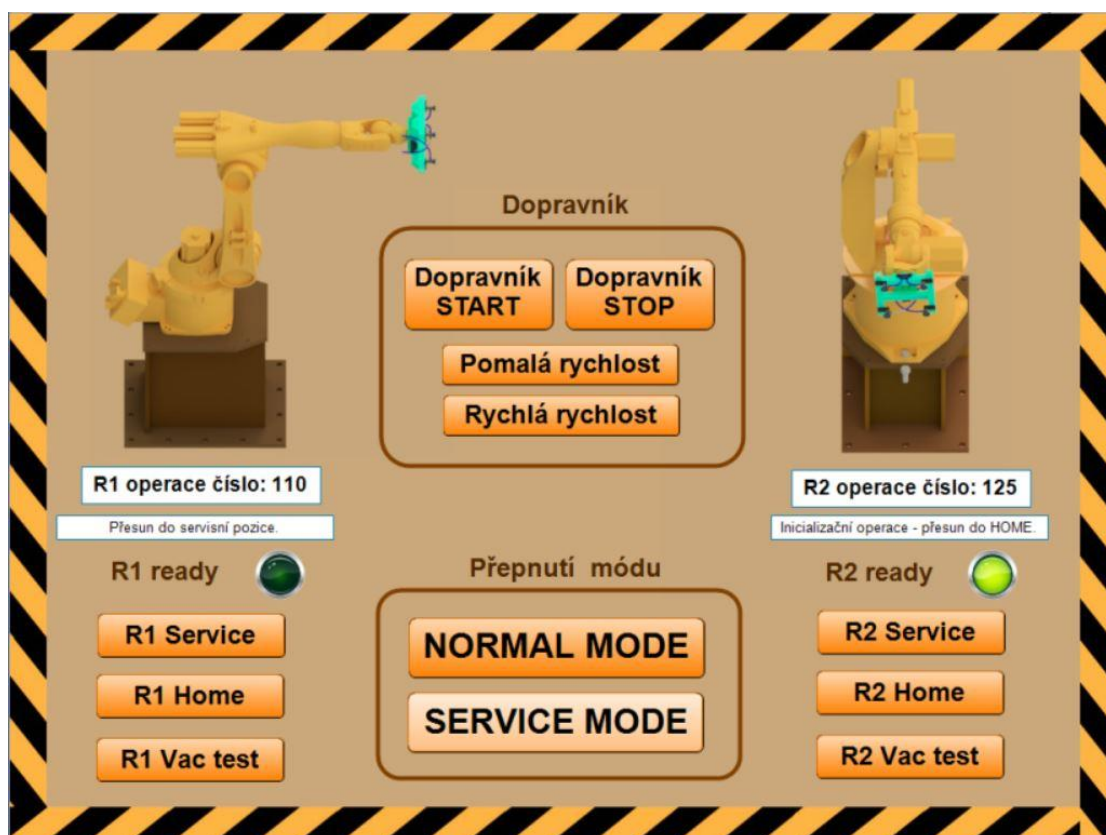
❖ Servisní mód

Pro funkci diagnostiky a údržby jsem vytvořil vizualizaci pro tzv. servisní mód, viz Obrázek 51. Obrazovka slouží pro testování funkčnosti dopravníku a obou robotů. Dále je vybavena pomocnými funkcemi pro údržbu zejména týkající se koncového efektoru. Tlačítka plní funkci:

- Service — přesun robota do polohy umožňující snadnou údržbu efektoru,
- Home — přesun robota do *Home* pozice,
- Vac test — testování funkčnosti pneumatických prvků související s efektozem,
- Dopravník START — aktivace pohybu dopravníku,
- Dopravník STOP — deaktivace pohybu dopravníku,
- Pomalá rychlost — testování rychlosti dopravníku, jež určená pro případ, kdy se na páse nacházejí dvě a méně součástí,
- Rychlá rychlost — testování rychlosti dopravníku, jež je určená pro případ, kdy se na páse nacházejí tři a více součástí,



Obr. 50) Vizualizace *NORMAL MODE*



Obr. 51) Vizualizace *SERVICE MODE*

8 ZHODNOCENÍ A DISKUZE

Cílem práce bylo především navrhnout koncepci pracoviště pro manipulační operace, které bude určeno výhradně pro experimentální činnost a výuku studentů. Pracoviště nebude oploceno. Bude obsahovat pouze základní bezpečnostní prvky jako je safety stop a bezpečnostní optická závora (SICK C40S). Důraz byl kladen na vypracování simulačního modelu v PS s ohledem k tzv. virtuálnímu zprovoznění na bázi externího připojení softPLC Beckhoff pomocí standardizovaného komunikačního protokolu OPC UA. Tudiž očekávaným výstupem byly zejména programy pro PLC a oba roboty, jež lze uplatnit do reálného pracoviště, a ověřený layout. Z těchto důvodů není relevantní se v práci věnovat problematice bezpečnosti, resp. analýze rizik (podle ČSN EN ISO 12100 apod.), která se v praxi běžně aplikuje na robotická pracoviště do reálného provozu.

Práce obsahuje nad rámec svého zadání podrobný návod pro tzv. virtuální zprovoznění, jež je uskutečněno propojením PS a softPLC Beckhoff prostřednictvím OPC UA serveru. Návod obsahuje základní kroky, které lze uplatnit na virtuální zprovoznění pracovišť jiného zaměření za předpokladu použití stejných softwarových nástrojů. Tudiž lze diplomová práce použít jako podklad pro navazující práce.

Ke snadnější demonstraci logiky, určené pro virtuální zprovoznění, byly nad rámec zadání vytvořeny vizualizace sloužící jako ovládací panel robotického pracoviště. Díky uživatelskému prostředí je možno snadno sledovat průběh dění během simulace. Případně lze pomocí panelu vyvolat neočekávanou událost a sledovat reakci systému.

9 ZÁVĚR

Cílem diplomové práce bylo navrhnout a následně tzv. virtuálně zprovoznit robotické experimentální pracoviště pro manipulační operace v prostředí softwaru Tecnomatix Siemens Process Simulate. Jedná se o robotickou buňku určenou pro edukativní a demonstrační účely, tudíž minimalizace délky trvání jednoho cyklu nebyl stěžejní úkol. Projekt se naopak snaží vytvořit robotické operace s co možná největším počtem různých interakcí ve smyslu robot – periferie. Do návrhu bylo nutné implementovat dostupné dílčí komponenty: dopravník DP50 2000x350, prototypové přípravky – stůl a skluz, robot KR 5 ARC a robot KR 16-2. Maximální rozměr pracoviště je dovolen v rozměrech 3800 x 4700 mm.

V počátku diplomové práce jsou uvedeny nejznámější alternativy k softwaru PS, jež disponují podobnými funkcemi. Lze konstatovat, že uvedený projekt lze pravděpodobně vypracovat ve všech softwarech uvedených v kapitole 3. Nicméně by bylo nutné si ověřit dostupnost robotických modelů včetně jejich kontrolérů pro řádnou implementaci.

Pátá kapitola obsahuje koncepční návrhy pro daný typ zadání, z nichž byl následně vybrán návrh číslo tři. Jednotlivé dílčí periferie jsou od sebe rozumně vzdáleny a nepřesahují hranice dovoleného prostoru pro celé pracoviště. Demonstrace nejrozličnějších typů interakcí ve vybrané koncepční variantě je splněna.

V diplomové práci byl navržen prototypový efektor. Jeho konstrukce se skládá převážně z modulárních dílů firmy Aluteck a pneumatických prvků společnosti Festo. Měnitelná rozteč přísavky v jednom směru zajišťuje variabilitu pro případnou záměnu manipulované součásti. Jako nejvhodnější efektor je k efektoru zvolen VADMI-95-LS-P, jenž vlastní druhý integrovaný elektromagnetický ventil pro vyfukovací impuls.

Vytvoření modelu pracovní buňky v PS proběhlo úspěšně, viz kapitola 6 s popisem jednotlivých kroků. Následně vytvořený automatický cyklus pro manipulaci součástí obsahuje celkem tři robotické operace (OP 20, OP 30 a OP 40), pohyb po dopravníku a „sklouznutí“ po skluzu. Celá simulace probíhá v nekonečné smyčce, o níž se starají dvě operace pro neustálé generování a odvod součástí z robotického pracoviště.

Celé řízení robotické buňky obstarává soft-PLC společnosti Beckhoff, jež je vytvořeno v programu TwinCAT 3 běžící na jádře Microsoft Visual Studio 2013. Pomocí jeho doplňku (TF6100 TC3 OPC UA) se podařilo navázat komunikaci se simulovaným modelem v programu PS. Komunikace, resp. řízení probíhá pomocí 34 signálů uvedených v Tabulce 4. Struktura PLC programu je rozepsána od kapitoly 7.4.4. Kromě automatického cyklu obsahuje funkce pro start, pozastavení a zastavení robotické buňky. V případě údržby je připravena sada funkcí pro servisní činnost.

Jelikož je třeba automatizované pracoviště i ovládat, vytvořil jsem sadu vizualizací, jež jsou optimalizovány pro 15" dotykový monitor s rozlišením 1024 x 768. Obrazovka tzv. normálního módu slouží k získání informací o aktuálním stavu pracoviště v reálném čase a pro ovládání automatického cyklu. Tzv. servisní mód umožňuje vyvolat robotické operace pro snadnější údržbu (zejména efektoru). Obsahuje navíc testovací funkce pro ověření funkčnosti jednotlivých periférií.

Virtuální simulace proběhla ve všech směrech úspěšně. Proto lze nasadit vytvořené programy do fyzického pracoviště postaveného dle ověřeného layoutu. Jedná se o

programy pro oba roboty KUKA v nativním jazyce KRL, PLC program ve strukturovaném textu a vizualizace pro ovládací panel. Ukázalo se, že tato kombinace může pro praxi představovat levný nástroj pro virtuální zprovoznění, jelikož je zapotřebí pouze jednoho osobního počítače.

10 SEZNAM POUŽITÝCH ZDROJŮ [20]

- [1] GURGUL, Michał, 2015. Robotics: Virtual Commissioning in Process Simulate. In: *Slideshare* [online]. LinkedIn [cit. 2018-05-25]. Dostupné z: <https://www.slideshare.net/mgurgul/robotics-virtual-comisionning-in-process-simulate>
- [2] Návod pro RobotStudio, c2018. In: *ABB* [online]. ABB [cit. 2018-05-25]. Dostupné z: <https://new.abb.com/products/robotics/cs/robotstudio/navody>
- [3] Virtuální zprovoznění: Od nápadu ke skutečnému stroji o třetinu rychleji, c2015. *Siemens* [online]. Praha: Siemens [cit. 2018-05-25]. Dostupné z: https://w5.siemens.com/web/cz/cz/corporate/portal/home/o_nas/veletrhy/msv/Documents/FS_virtual_commissioning_January_2015_CZ.PDF
- [4] CEJNAROVÁ, Andrea, 2015. Od 1. průmyslové revoluce ke 4. *Technický týdeník* [online]. Praha: Business Media CZ, 4. červen 2015, 1 [cit. 2018-05-25]. Dostupné z: https://www.technickytydenik.cz/rubriky/ekonomika-byznys/od-1-prumyslove-revoluce-ke-4_31001.html
- [5] DELMIA Robotics, c2014. *Digital Manufacturing Solutions* [online]. DMSCO.COM [cit. 2018-05-25]. Dostupné z: <http://4dmsco.com/solutions/delmia/delmia-robotics/>
- [6] *Robotics virtual commissioning: Datasheet* [online], Dassault Systemes, 2 [cit. 2018-05-25]. Dostupné z: https://www.3ds.com/fileadmin/PRODUCTS/DELMIA/PDF/DM-12877-Robotics-Virtual-Commissioning-Datasheet_HR.pdf
- [7] *RobotStudio™ 6: Průmyslový software* [online], ABB, 2 [cit. 2018-05-25]. Dostupné z: <http://search.abb.com/library/Download.aspx?DocumentID=9AKK107046A1349&LanguageCode=cs&DocumentPartId=&Action=Launch>
- [8] RobotStudio, ABB [online]. ABB [cit. 2018-05-25]. Dostupné z: <https://new.abb.com/products/robotics/cs/robotstudio>
- [9] JUCKES, John, *JT OPEN: XT B-REP; Making it Real* [online]. 12. září 2010, 23 [cit. 2018-05-25]. Dostupné z: https://www.plm.automation.siemens.com/ja_jp/Images/XT-in-JT_tcm821-115289.pdf
- [10] Tecnomatix eMServer Platform 13.0 Help. *PLM Automation* [online]. SRN: SIEMENS, 2016 [cit. 2018-05-25]. Dostupné z: https://docs.plm.automation.siemens.com/tdoc/tecnomatix/13/tecnomatix_eMS/
- [11] Tecnomatix Process Simulate. *Digital Factory* [online]. Plzeň: Západočeská univerzita v Plzni, 2013 [cit. 2018-05-25]. Dostupné z: <https://www.digipod.zcu.cz/index.php/oblasti-nasazeni/robotika/tecnomatix-process-simulate>
- [12] AXIOM TECH S. R. O. Virtual Commissioning pomáhá navrhovat robotické linky. *Automa* [online]. Děčín: Automa-časopis pro automatizační techniku, 2016, **21**(5), 2 [cit. 2018-05-25]. ISSN 1210-9592. Dostupné z: http://automa.cz/Aton/FileRepository/pdf_articles/54805.pdf

- [13] NUÑEZ, angelica daniela, 06 Delmia Robotics Posicionamiento de grippers en robots. In: *YouTube* [online]. 17.6.2012 [cit. 2018-05-25]. Dostupné z: https://youtu.be/WizfJwvzn_E
- [14] KR 16-2: Specification, 2011. *Robolution* [online]. KUKA Roboter [cit. 2018-05-25]. Dostupné z: <http://robolution.eu/letoltes.php?type=media&name=960-kuka-kr-16-2-robot-adatlap.pdf>
- [15] KR 5 arc: Specification, c2016. *KUKA* [online]. KUKA Roboter [cit. 2018-05-25]. Dostupné z: https://www.kuka.com/-/media/kuka-downloads/imported/48ec812b1b2947898ac2598aff70abc0/spez_kr_5_arc_en.pdf
- [16] Vakuové přísavky ESG, 2016. *Festo* [online]. Festo, 2016 [cit. 2018-05-25]. Dostupné z: https://www.festo.com/cat/cs_cz/data/doc_cs/PDF/CZ/ESG_CZ.PDF
- [17] Beckhoff Information System, *Beckhoff: Infosys* [online]. Beckhoff Automation GmbH & Co. [cit. 2018-05-25]. Dostupné z: https://infosys.beckhoff.com/index_en.htm
- [18] Process Designer. *AXIOM TECH* [online]. Zlín: AXIOM TECH, 2016 [cit. 2018-05-25]. Dostupné z: <https://www.axiomtech.cz/24828-tecnomatix-process-designer>
- [19] VITÁK, Jan, 2016. *Simulátor výrobních linek*. Brno. Diplomová práce. Vysoké učení technické v Brně. Vedoucí práce Ing. Jan Pásek, CSc.
- [20] CITACE PRO. *Generátor citací* [online]. 2016 [cit. 2018-05-25]. Dostupné z: <http://citace.lib.vutbr.cz/info>

11 SEZNAM ZKRATEK, SYMBOLŮ, OBRÁZKŮ A TABULEK

11.1 Seznam tabulek

Tab 1)	Základní parametry dopravníku, pohonu a převodovky.....	18
Tab 2)	Základní parametry robotů [14, 15].....	21
Tab 3)	Objemy jednotlivých komponent [16].....	33
Tab 4)	Signály pro komunikaci PS s PLC	58

11.2 Seznam obrázků

Obr. 1)	Schématické propojení jednotlivých softwarových nástrojů.....	13
Obr. 2)	Softwarové nástroje pro digitální vývoj průmyslu [13, 2].....	16
Obr. 3)	Rozměry manipulované součásti	17
Obr. 4)	Dopravník DP50 2000x350	19
Obr. 5)	Kinematické schéma pohonu dopravníku.....	19
Obr. 6)	Dostupné přípravky	20
Obr. 7)	Modely robotů KUKA	21
Obr. 8)	Nevyklizený pracovní prostor v hale.....	22
Obr. 9)	Základní rozměry pracovního prostoru	22
Obr. 10)	Varianta 1 – koncepční layout	23
Obr. 11)	Varianta 2 – koncepční layout	24
Obr. 12)	Varianta 3 – koncepční layout	25
Obr. 13)	Nastavení souboru pro správný export do *.JT	28
Obr. 14)	Vytvoření .cojt „souboru“.....	28
Obr. 15)	Definování typu komponent v zadaném projektu.....	29
Obr. 16)	Funkce pro přesun objektů	29
Obr. 17)	Měnitelná rozteč efektoru	31
Obr. 18)	Nejrizikovější sklon efektoru.....	31
Obr. 19)	Pneumatický systém manipulačního robota	33
Obr. 20)	Doba odsátí jednoho litru vzduchu dle typu ejektoru [16]	34
Obr. 21)	Definování nástroje v PS	35
Obr. 22)	Dialogové okno pro připevnění efektoru k robotu	36
Obr. 23)	Definování senzoru v PS	37
Obr. 24)	Trajektorie dopravníku	38
Obr. 25)	Vytvoření signálů pro ovládání dopravníku	38
Obr. 26)	Řídící signály robota.....	39
Obr. 27)	Změna Home pozice robota	40
Obr. 28)	Dialogové okno <i>Snapshot Editor</i>	41
Obr. 29)	Pick and Place operace	42
Obr. 30)	Úprava lokací.....	43
Obr. 31)	Dialogové okno <i>Robotic Program Inventory</i>	44
Obr. 32)	Ukázka nastavené operace 40 v <i>Path Editor</i>	45
Obr. 33)	Přřazení součásti k operaci	46

Obr. 34)	Definování materiálového toku	47
Obr. 35)	Rozdíl ve vykonání stejného procesu ve dvou režimech PS [1]	49
Obr. 36)	Prvky OPC UA instance v tray liště	50
Obr. 37)	Výchozí nastavení připojení k OPC serveru	51
Obr. 38)	Spuštění SW TwinCAT	51
Obr. 39)	Prodloužení licencí v SW TwinCAT	52
Obr. 40)	Ukázka deklarace proměnných v souboru <i>GVL</i>	53
Obr. 41)	Karta <i>PLC</i> v dialogovém okně <i>Options</i>	54
Obr. 42)	Dialogové okno <i>External Connections</i>	54
Obr. 43)	Signály připravené na import	55
Obr. 44)	Znázornění operací v pracovišti	56
Obr. 45)	Umístění senzorů na pracovišti	57
Obr. 46)	Hlavní skupiny procesů PLC programu	59
Obr. 47)	Kolizní zóna	60
Obr. 48)	Flowchart k řízení dopravníku	61
Obr. 49)	Flowchart k řízení robotů	62
Obr. 50)	Vizualizace <i>NORMAL MODE</i>	64
Obr. 51)	Vizualizace <i>SERVICE MODE</i>	64

11.3 Seznam použitých symbolů

SYMBOL	LEGENDA	VELIČINA
a	Zrychlení přísavky, resp. robota	$[\text{m} \cdot \text{sec}^{-2}]$
D_p	Průměr přísavky	$[\text{mm}]$
D_v	Průměr hnacího válce	$[\text{mm}]$
f	Koeficient tření	$[-]$
F_a	Reakční síla od zrychlení	$[\text{N}]$
F_g	Tíhová síla	$[\text{N}]$
F_p	Přidržná síla	$[\text{N}]$
F_{pj}	Přidržná síla pro jednu přísavku	$[\text{N}]$
F_{px}	Část přidržné síly (osa x)	$[\text{N}]$
F_{py}	Část přidržné síly (osa y)	$[\text{N}]$
F_t	Třecí síla	$[\text{N}]$
g	Gravitační konstanta	$[\text{m} \cdot \text{sec}^{-2}]$
i	Převodový poměr	$[-]$
k	Součinitel bezpečnosti	$[-]$
L_{dd}	Délka dopravníku	$[\text{mm}]$
$L_{dš}$	Šířka dopravníku	$[\text{mm}]$
L_{dv}	Výška dopravníku	$[\text{mm}]$
m	Hmotnost součásti	$[\text{kg}]$

SYMBOL	LEGENDA	VELIČINA
M_j	Jmenovitý točivý moment	[Nm]
n	Počet přísavek na efektoru	[–]
n_j	Jmenovité otáčky motoru	$[\text{min}^{-1}]$
n_v	Jmenovité otáčky hnacího válce dopravníku	$[\text{min}^{-1}]$
t_{70}	Doba odsátí 1 litru vzduchu pomocí ejektoru VADMI–70	[sec]
t_{95}	doba odsátí 1 litru vzduchu pomocí ejektoru VADMI–95	[sec]
t_e	Doporučená doba odsátí vzduchu	[sec]
t_{e70}	Doba odsátí 1 litru vzduchu o objemu „ V_c “ pomocí ejektoru VADMI–70	[sec]
t_{e95}	Doba odsátí 1 litru vzduchu o objemu „ V_c “ pomocí ejektoru VADMI–95	[sec]
V_1	Objem vzduchu přísavky s držákem	$[\text{cm}^3]$
V_2	Objem vzduchu T–spojky	$[\text{cm}^3]$
V_3	Objem vzduchu rozdělovače	$[\text{cm}^3]$
V_4	Objem vzduchu hadice (přísavka - rozdělovač)	$[\text{cm}^3]$
V_5	Objem vzduchu hadice (rozdělovač - T–spojka)	$[\text{cm}^3]$
V_6	Objem vzduchu přísavky	$[\text{cm}^3]$
V_c	Celkový objem vzduchu	$[\text{cm}^3]$
v_d	Rychlost pásu dopravníku	$[\text{mm} \cdot \text{sec}^{-1}]$
V_p	Objem vzduchu přísavky	$[\text{cm}^3]$
Δp_u	Změna tlaku v pneumatické soustavě	[bar]

SEZNAM PŘÍLOH

PŘÍLOHA 1 Festo – výběr přísavek

VÝKRESY

DP – 162007 – 12 STŘEDÍCÍ VÁLEC

DP – 162007 – 11 PŘÍRUBA

DP – 162007 – 10 ÚCHYT

DP – 162007 – 1 – 1/2 SESTAVA EFEKTOR

DP – 162007 – 1 – 2/2 MONTÁŽNÁ SESTAVA EFEKTOR

DP – 162007 – 0 LAYOUT PRACOVIŠTĚ

PŘÍLOHA 1

Basic principles of vacuum technology

Introduction

FESTO

Step 3: Calculating the holding force and breakaway force

Determining the breakaway force

F_A = Theoretical breakaway force
[N]

F_H = Theoretical holding force of the
suction gripper [N]
(Result → 45)

n = Number of suction grippers
(2 suction grippers are
planned in the problem
example)






$$F_A = \frac{F_H}{n}$$

Example:

$$F_A = \frac{93 \text{ N}}{2}$$

$$F_A \approx 47 \text{ N}$$

Breakaway force F_A dependent on suction cup diameter and suction cup shape

Round suction cup		F_A at -0.7 bar				Oval suction cup		F_A at -0.7 bar
Ordering data	Suction cup \varnothing [mm]	Standard 	Extra deep 	Bellows, 1.5 convol- utions 	Bellows, 3.5 convol- utions 	Ordering data	Suction cup size [mm]	Oval 
→ ess	2	0.1 N				→ ess	4x10	2 N
	4	0.4 N					4x20	3.4 N
	6	1.1 N					6x10	2.9 N
	8	2.3 N					6x20	5.9 N
	10	3.9 N					8x20	8 N
	15	8.5 N					8x30	10.9 N
	20	16.3 N	9.8 N				10x30	15.2 N
	30	40.8 N	37.2 N	12.9 N	8.2 N		15x45	32 N
	40	69.6 N	67.6 N	52.3 N	42.4 N		20x60	62.8 N
	50	105.8 N	103.6 N	72.6 N	63.4 N		25x75	92.5 N
	60	166.1 N	162.5 N				30x90	134.4 N
	80	309.7 N	275 N	213.9 N				
	100	503.6 N	440.8 N					
	150	900 N						
	200	1,610 N						

Breakaway force F_A too low



Reliable range
for the problem example



Suction cup diameter too big for
workpiece

In this example we opt for 2 suction
grippers:

- Round design
- Suction cup diameter 40 mm
- Breakaway force of 69.6 N



Note

The load capacity of the vacuum
suction gripper must be greater
than the calculated value.